User's Manual

32

R32C/102 Group

User's Manual: Hardware

RENESAS MCU

M16C Family / R32C/100 Series

All information contained in these materials, including products and product specifications, represents information on the product at the time of publication and is subject to change by Renesas Electronics Corp. without notice. Please review the latest informaton published by Renesas Electronics Corp. through various means, including the Renesas Electronics Corp. website (http://www.renesas.com).

Renesas Electronics
www.renesas.com

Rev. 1.01 Nov 2010

Notice

1. All information included in this document is current as of the date this document is issued. Such information, however, is subject to change without any prior notice. Before purchasing or using any Renesas Electronics products listed herein, please confirm the latest product information with a Renesas Electronics sales office. Also, please pay regular and careful attention to additional and different information to be disclosed by Renesas Electronics such as that disclosed through our website.
2. Renesas Electronics does not assume any liability for infringement of patents, copyrights, or other intellectual property rights of third parties by or arising from the use of Renesas Electronics products or technical information described in this document. No license, express, implied or otherwise, is granted hereby under any patents, copyrights or other intellectual property rights of Renesas Electronics or others.
3. You should not alter, modify, copy, or otherwise misappropriate any Renesas Electronics product, whether in whole or in part.
4. Descriptions of circuits, software and other related information in this document are provided only to illustrate the operation of semiconductor products and application examples. You are fully responsible for the incorporation of these circuits, software, and information in the design of your equipment. Renesas Electronics assumes no responsibility for any losses incurred by you or third parties arising from the use of these circuits, software, or information.
5. When exporting the products or technology described in this document, you should comply with the applicable export control laws and regulations and follow the procedures required by such laws and regulations. You should not use Renesas Electronics products or the technology described in this document for any purpose relating to military applications or use by the military, including but not limited to the development of weapons of mass destruction. Renesas Electronics products and technology may not be used for or incorporated into any products or systems whose manufacture, use, or sale is prohibited under any applicable domestic or foreign laws or regulations.
6. Renesas Electronics has used reasonable care in preparing the information included in this document, but Renesas Electronics does not warrant that such information is error free. Renesas Electronics assumes no liability whatsoever for any damages incurred by you resulting from errors in or omissions from the information included herein.
7. Renesas Electronics products are classified according to the following three quality grades: "Standard", "High Quality", and "Specific". The recommended applications for each Renesas Electronics product depends on the product's quality grade, as indicated below. You must check the quality grade of each Renesas Electronics product before using it in a particular application. You may not use any Renesas Electronics product for any application categorized as "Specific" without the prior written consent of Renesas Electronics. Further, you may not use any Renesas Electronics product for any application for which it is not intended without the prior written consent of Renesas Electronics. Renesas Electronics shall not be in any way liable for any damages or losses incurred by you or third parties arising from the use of any Renesas Electronics product for an application categorized as "Specific" or for which the product is not intended where you have failed to obtain the prior written consent of Renesas Electronics. The quality grade of each Renesas Electronics product is "Standard" unless otherwise expressly specified in a Renesas Electronics data sheets or data books, etc.

   "Standard": Computers; office equipment; communications equipment; test and measurement equipment; audio and visual equipment; home electronic appliances; machine tools; personal electronic equipment; and industrial robots.

   "High Quality": Transportation equipment (automobiles, trains, ships, etc.); traffic control systems; anti-disaster systems; anti-crime systems; safety equipment; and medical equipment not specifically designed for life support.

   "Specific": Aircraft; aerospace equipment; submersible repeaters; nuclear reactor control systems; medical equipment or systems for life support (e.g. artificial life support devices or systems), surgical implantations, or healthcare intervention (e.g. excision, etc.), and any other applications or purposes that pose a direct threat to human life.
8. You should use the Renesas Electronics products described in this document within the range specified by Renesas Electronics, especially with respect to the maximum rating, operating supply voltage range, movement power voltage range, heat radiation characteristics, installation and other product characteristics. Renesas Electronics shall have no liability for malfunctions or damages arising out of the use of Renesas Electronics products beyond such specified ranges.
9. Although Renesas Electronics endeavors to improve the quality and reliability of its products, semiconductor products have specific characteristics such as the occurrence of failure at a certain rate and malfunctions under certain use conditions. Further, Renesas Electronics products are not subject to radiation resistance design. Please be sure to implement safety measures to guard them against the possibility of physical injury, and injury or damage caused by fire in the event of the failure of a Renesas Electronics product, such as safety design for hardware and software including but not limited to redundancy, fire control and malfunction prevention, appropriate treatment for aging degradation or any other appropriate measures. Because the evaluation of microcomputer software alone is very difficult, please evaluate the safety of the final products or system manufactured by you.
10. Please contact a Renesas Electronics sales office for details as to environmental matters such as the environmental compatibility of each Renesas Electronics product. Please use Renesas Electronics products in compliance with all applicable laws and regulations that regulate the inclusion or use of controlled substances, including without limitation, the EU RoHS Directive. Renesas Electronics assumes no liability for damages or losses occurring as a result of your noncompliance with applicable laws and regulations.
11. This document may not be reproduced or duplicated, in any form, in whole or in part, without prior written consent of Renesas Electronics.
12. Please contact a Renesas Electronics sales office if you have any questions regarding the information contained in this document or Renesas Electronics products, or if you have any other inquiries.

(Note 1) "Renesas Electronics" as used in this document means Renesas Electronics Corporation and also includes its majority-owned subsidiaries.

(Note 2) "Renesas Electronics product(s)" means any product developed or manufactured by or for Renesas Electronics.

General Precautions in the Handling of MPU/MCU Products

The following usage notes are applicable to all MPU/MCU products from Renesas. For detailed usage notes on the products covered by this manual, refer to the relevant sections of the manual. If the descriptions under General Precautions in the Handling of MPU/MCU Products and in the body of the manual differ from each other, the description in the body of the manual takes precedence.

1. Handling of Unused Pins

   Handle unused pins in accord with the directions given under Handling of Unused Pins in the manual.

   — The input pins of CMOS products are generally in the high-impedance state. In operation with an unused pin in the open-circuit state, extra electromagnetic noise is induced in the vicinity of LSI, an associated shoot-through current flows internally, and malfunctions occur due to the false recognition of the pin state as an input signal become possible. Unused pins should be handled as described under Handling of Unused Pins in the manual.

2. Processing at Power-on

   The state of the product is undefined at the moment when power is supplied.

   — The states of internal circuits in the LSI are indeterminate and the states of register settings and pins are undefined at the moment when power is supplied.

   In a finished product where the reset signal is applied to the external reset pin, the states of pins are not guaranteed from the moment when power is supplied until the reset process is completed.

   In a similar way, the states of pins in a product that is reset by an on-chip power-on reset function are not guaranteed from the moment when power is supplied until the power reaches the level at which resetting has been specified.

3. Prohibition of Access to Reserved Addresses

   Access to reserved addresses is prohibited.

   — The reserved addresses are provided for the possible future expansion of functions. Do not access these addresses; the correct operation of LSI is not guaranteed if they are accessed.

4. Clock Signals

   After applying a reset, only release the reset line after the operating clock signal has become stable. When switching the clock signal during program execution, wait until the target clock signal has stabilized.

   — When the clock signal is generated with an external resonator (or from an external oscillator) during a reset, ensure that the reset line is only released after full stabilization of the clock signal. Moreover, when switching to a clock signal produced with an external resonator (or by an external oscillator) while program execution is in progress, wait until the target clock signal is stable.

5. Differences between Products

   Before changing from one product to another, i.e. to one with a different part number, confirm that the change will not lead to problems.

   — The characteristics of MPU/MCU in the same group but having different part numbers may differ because of the differences in internal memory capacity and layout pattern. When changing to products of different part numbers, implement a system-evaluation test for each of the products.

About This Manual

1. Purpose and Target User

This manual is designed to be read primarily by application developers who have an understanding of this microcomputer (MCU) including its hardware functions and electrical characteristics. The user should have a basic understanding of electric circuits, logic circuits and, MCUs.
This manual consists of 29 chapters covering six main categories: Overview, CPU, System Control, Peripherals, Electrical Characteristics, and Usage Notes.

Carefully read all notes in this document prior to use. Notes are found throughout each chapter, at the end of each chapter, and in the dedicated Usage Notes chapter.

The Revision History at the end of this manual summarizes primary modifications and additions to the previous versions. For details, please refer to the relative chapters or sections of this manual.

The R32C/102 Group includes the documents listed below. Verify this manual is the latest version by visiting the Renesas Electronics website.

Type of Document	Contents	Document Name	Document Number
Datasheet	Overview of Hardware and Electrical Characteristics	R32C/102 Group Datasheet	REJ03B0300-0101
User's Manual: Hardware	Specifications and detailed descriptions of: -pin layout -memory map -peripherals -electrical characteristics -timing characteristics Refer to the Application Manual for peripheral usage.	R32C/102 Group User's Manual: Hardware	This publication
User's Manual: Software/Software Manual	Descriptions of instruction set	R32C/100 Series Software Manual	REJ09B0267-0100
Application Note	-Usages -Applications -Sample programs -Programing technics using Assembly language or C programming language	Available on the Renesas Electronics website.	
Renesas Technical Update	Bulletins on product specifications, documents, etc.		

2. Numbers and Symbols

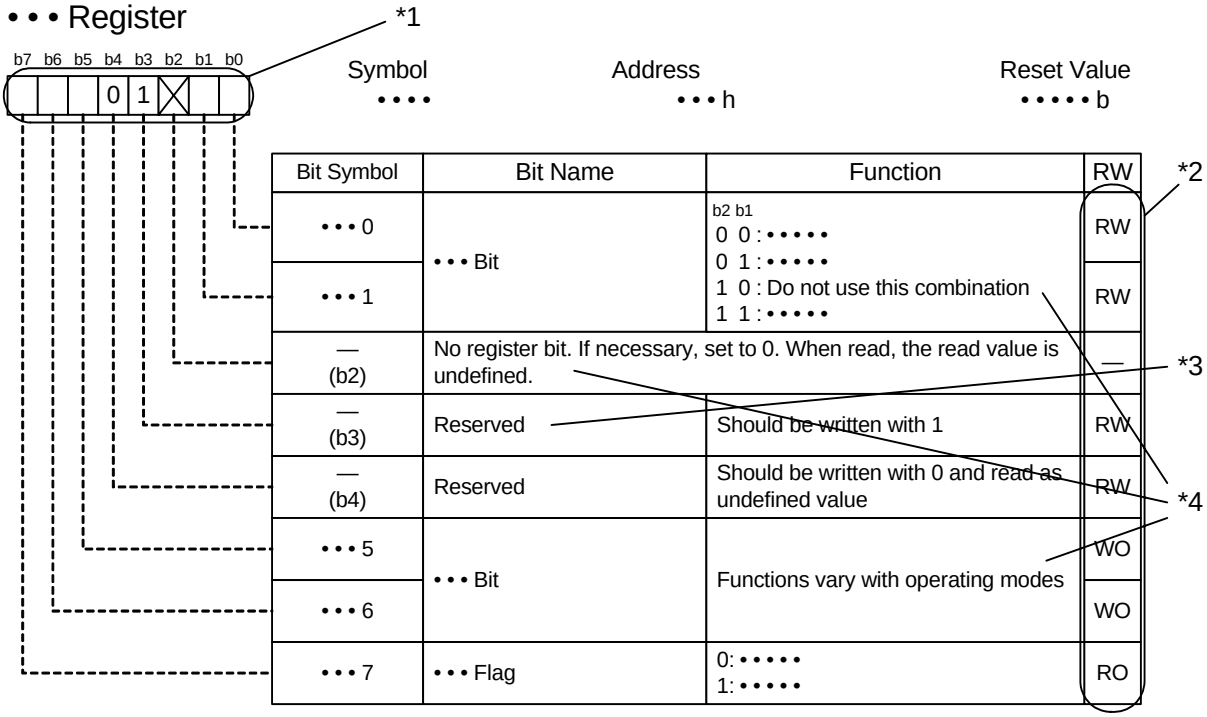
The following explains the denotations used in this manual for registers, bits, pins and various numbers.

(1) Registers, bits, and pins
Registers, bits, and pins are indicated by symbols. Each symbol has a register/bit/pin identifier after the symbol.
Example: PM03 bit in the PM0 register
P3_5 pin, VCC pin

(2) Numbers
A binary number has the suffix “b” except for a 1-bit value.
A hexadecimal number has the suffix “h”.
A decimal number has no suffix.
Example: Binary notation: 11b
Hexadecimal notation: EFA0h
Decimal notation: 1234

3. Registers

The following illustration describes registers used throughout this manual.



*1
Blank box: Set this bit to 0 or 1 according to the function.
0: Set this bit to 0.
1: Set this bit to 1.
X: Nothing is assigned to this bit.

*2
RW: Read and write
RO: Read only
WO: Write only (the read value is undefined)
—: Not applicable

*3
- Reserved bit: This bit field is reserved. Set this bit to a specified value. For RW bits, the written value is read unless otherwise noted.

*4
- No register bit(s): No register bit(s) is/are assigned to this field. If necessary, set to 0 for possible future implementation.
- Do not use this combination: Proper operation is not guaranteed when this value is set.
- Functions vary with operating modes: Functions vary with peripheral operating modes. Refer to register illustrations of the respective mode.

4. Abbreviations and Acronyms

The following acronyms and terms are used throughout this manual.

Abbreviation/Acronym	Meaning
ACIA	Asynchronous Communication Interface Adapter
bps	bits per second
CRC	Cyclic Redundancy Check
DMA	Direct Memory Access
DMAC	Direct Memory Access Controller
GSM	Global System for Mobile Communications
Hi-Z	High Impedance
IEBus	Inter Equipment Bus
I/O	Input/Output
IrDA	Infrared Data Association
LSB	Least Significant Bit
MSB	Most Significant Bit
NC	Non-Connection
PLL	Phase Locked Loop
PWM	Pulse Width Modulation
SIM	Subscriber Identity Module
UART	Universal Asynchronous Receiver/Transmitter
VCO	Voltage Controlled Oscillator

All trademarks and registered trademarks are the property of their respective owners.

TABLE OF CONTENTS

1. Overview 1

1.1 Features ... 1

1.1.1 Applications ... 1

1.1.2 Performance Overview ... 2

1.2 Product Information ... 4

1.3 Block Diagram ... 5

1.4 Pin Assignments ... 6

1.5 Pin Definitions and Functions ... 12

2. Central Processing Unit (CPU) 16

2.1 General Purpose Registers ... 17

2.1.1 Data Registers (R2R0, R3R1, R6R4, and R7R5) ... 17

2.1.2 Address Registers (A0, A1, A2, and A3) ... 17

2.1.3 Static Base Register (SB) ... 17

2.1.4 Frame Base Register (FB) ... 17

2.1.5 Program Counter (PC) ... 17

2.1.6 Interrupt Vector Table Base Register (INTB) ... 17

2.1.7 User Stack Pointer (USP) and Interrupt Stack Pointer (ISP) ... 17

2.1.8 Flag Register (FLG) ... 17

2.2 Fast Interrupt Registers ... 19

2.2.1 Save Flag Register (SVF) ... 19

2.2.2 Save PC Register (SVP) ... 19

2.2.3 Vector Register (VCT) ... 19

2.3 DMAC-associated Registers ... 19

2.3.1 DMA Mode Registers (DMD0, DMD1, DMD2, and DMD3) ... 19

2.3.2 DMA Terminal Count Registers (DCT0, DCT1, DCT2, and DCT3) ... 19

2.3.3 DMA Terminal Count Reload Registers (DCR0, DCR1, DCR2, and DCR3) ... 19

2.3.4 DMA Source Address Registers (DSA0, DSA1, DSA2, and DSA3) ... 19

2.3.5 DMA Source Address Reload Registers (DSR0, DSR1, DSR2, and DSR3) ... 19

2.3.6 DMA Destination Address Registers (DDA0, DDA1, DDA2, and DDA3) ... 19

2.3.7 DMA Destination Address Reload Registers (DDR0, DDR1, DDR2, and DDR3) ... 19

3. Memory 20

4. Special Function Registers (SFRs) 21

5. Resets 60

5.1 Hardware Reset ... 60

5.2 Software Reset ... 62

5.3 Watchdog Timer Reset ... 62

5.4 Reset Vector ... 63

6. Power Management 64
6.1 Voltage Regulators for Internal Logic ... 64
6.1.1 Decoupling Capacitor ... 65
7. Clock Generator 66
7.1 Clock Generator Types ... 66
7.1.1 Main Clock ... 75
7.1.2 Sub Clock (fC) ... 76
7.1.3 PLL Clock ... 77
7.1.4 On-chip Oscillator Clock ... 79
7.2 Oscillator Stop Detection ... 80
7.2.1 How to Use Oscillator Stop Detection ... 80
7.3 Base Clock ... 80
7.4 CPU Clock and Peripheral Bus Clock ... 81
7.5 Peripheral Clock ... 81
7.6 Clock Output Function ... 82
7.7 Power Control ... 83
7.7.1 Normal Operating Mode ... 84
7.7.2 Wait Mode ... 89
7.7.3 Stop Mode ... 92
7.8 System Clock Protection ... 94
7.9 Notes on Clock Generator ... 95
7.9.1 Sub Clock ... 95
7.9.2 Power Control ... 95
8. Bus 96
8.1 Bus Setting ... 96
8.2 Peripheral Bus Timing Setting ... 97
9. Protection 98
9.1 Protect Register (PRCR Register) ... 98
9.2 Protect Register 2 (PRCR2 Register) ... 99
9.3 Protect Register 3 (PRCR3 Register) ... 99
9.4 Protect Release Register (PRR Register) ... 100
10. Interrupts 101
10.1 Interrupt Types ... 101
10.2 Software Interrupt ... 102
10.3 Hardware Interrupt ... 103
10.3.1 Special Interrupt ... 103
10.3.2 Peripheral Interrupt ... 103
10.4 Fast Interrupt ... 104
10.5 Interrupt Vectors ... 104

10.5.1 Fixed Vector Table 105
10.5.2 Relocatable Vector Table 105
10.6 Interrupt Request Acceptance 110
10.6.1 I Flag and IPL 110
10.6.2 Interrupt Control Register 111
10.6.3 Wake-up IPL Setting Register 114
10.6.4 Interrupt Sequence 115
10.6.5 Interrupt Response Time 116
10.6.6 IPL After Interrupt Request Acceptance 117
10.6.7 Register Saving 117
10.7 Register Restoring from Interrupt Handler 118
10.8 Interrupt Priority 118
10.9 Priority Resolver 118
10.10 External Interrupt 120
10.11 NMI 121
10.12 Key Input Interrupt 122
10.13 Intelligent I/O Interrupt 123
10.14 Notes on Interrupts 126
10.14.1 ISP Setting 126
10.14.2 NMI 126
10.14.3 External Interrupt 126

11. Watchdog Timer 127

12. DMAC 129
12.1 Transfer Cycle 138
12.1.1 Effect of Transfer Address and Data Bus Width 138
12.1.2 Effect of Bus Timing 139
12.2 DMA Transfer Cycle 141
12.3 Channel Priority and DMA Transfer Timing 142
12.4 Notes on DMAC 143
12.4.1 DMAC-associated Register Settings 143
12.4.2 Read from DMAC-associated Registers 143

13. DMAC II 144
13.1 DMAC II Settings 144
13.1.1 Registers RIPL1 and RIPL2 145
13.1.2 DMAC II Index 146
13.1.3 Interrupt Control Register of the Peripheral Function 149
13.1.4 Relocatable Vector Table of the Peripheral Function 149
13.1.5 IRLT Bit in the IIOiIE Register (i = 0 to 11) 149
13.2 DMAC II Performance 149

13.3 Transfer Types 149
13.3.1 Memory-to-memory Transfer 149
13.3.2 Immediate Data Transfer 150
13.3.3 Calculation Transfer 150
13.4 Transfer Modes 150
13.4.1 Single Transfer 150
13.4.2 Burst Transfer 150
13.4.3 Multiple Transfer 150
13.5 Chained Transfer 151
13.6 DMA II Transfer Complete Interrupt 151
13.7 Execution Time 152

14. Programmable I/O Ports 153
14.1 Port Pi Register (Pi register, i = 0 to 15) 155

15. Timers 156
15.1 Timer A 158
15.1.1 Timer Mode 165
15.1.2 Event Counter Mode 167
15.1.3 One-shot Timer Mode 171
15.1.4 Pulse-width Modulation Mode 173
15.2 Timer B 176
15.2.1 Timer Mode 179
15.2.2 Event Counter Mode 181
15.2.3 Pulse Period/Pulse-width Measure Mode 183
15.3 Notes on Timers 186
15.3.1 Timer A and Timer B 186
15.3.2 Timer A 186
15.3.3 Timer B 188

16. Three-phase Motor Control Timers 189
16.1 Modulation Modes of Three-phase Motor Control Timers 196
16.2 Timer B2 197
16.3 Timers A4, A1, and A2 199
16.4 Simultaneous Conduction Prevention and Dead Time Timer 202
16.5 Three-phase Motor Control Timer Operation 203
16.6 Notes on Three-phase Motor Control Timers 206
16.6.1 Shutdown 206
16.6.2 Register setting 206

17. Serial Interface 207
17.1 Synchronous Serial Interface Mode 224
17.1.1 Reset Procedure on Transmit/Receive Error 229

17.1.2 CLK Polarity ... 229
17.1.3 LSB First and MSB First Selection ... 230
17.1.4 Continuous Receive Mode ... 230
17.1.5 Serial Data Logical Inversion ... 231
17.1.6 CTS/RTS Function ... 231
17.2 Asynchronous Serial Interface Mode (UART Mode) ... 232
17.2.1 Bit Rate ... 237
17.2.2 Reset Procedure on Transmit/Receive Error ... 238
17.2.3 LSB First and MSB First Selection ... 238
17.2.4 Serial Data Logical Inversion ... 239
17.2.5 TXD and RXD I/O Polarity Inversion ... 240
17.2.6 CTS/RTS Function ... 240
17.3 Special Mode 1 (I^2C Mode) ... 241
17.3.1 Start Condition and Stop Condition Detection ... 247
17.3.2 Start Condition and Stop Condition Generation ... 247
17.3.3 Arbitration ... 248
17.3.4 SCL Control and Clock Synchronization ... 249
17.3.5 SDA Output ... 251
17.3.6 SDA Input ... 251
17.3.7 Acknowledge ... 251
17.3.8 Initialization of Transmit/Receive Operation ... 251
17.4 Special Mode 2 ... 252
17.4.1 $\overline{SSi}$ Input Pin Function (i = 0 to 6) ... 254
17.4.2 Clock Phase Setting ... 255
17.5 Notes on Serial Interface ... 257
17.5.1 Changing the UiBRG Register (i = 0 to 8) ... 257
17.5.2 Synchronous Serial Interface Mode ... 257
17.5.3 Special Mode 1 (I^2C Mode) ... 257
17.5.4 Reset Procedure on Communication Error ... 258
18. A/D Converter 259
18.1 Mode Descriptions ... 267
18.1.1 One-shot Mode ... 267
18.1.2 Repeat Mode ... 268
18.1.3 Single Sweep Mode ... 269
18.1.4 Repeat Sweep Mode 0 ... 270
18.1.5 Repeat Sweep Mode 1 ... 271
18.1.6 Multi-port Single Sweep Mode ... 272
18.1.7 Multi-port Repeat Sweep Mode 0 ... 273
18.2 Functions ... 274
18.2.1 Resolution Selection ... 274

18.2.2 Sample and Hold Function 274
18.2.3 Trigger Selection 274
18.2.4 DMAC Operating Mode 274
18.2.5 Function-extended Analog Input Pins 275
18.2.6 External Operating Amplifier (Op-AMP) Connection Mode 275
18.2.7 Power Saving 276
18.2.8 Output Impedance of Sensor Equivalent Circuit under A/D Conversion 276
18.3 Notes on A/D Converter 278
18.3.1 Notes on Designing Boards 278
18.3.2 Notes on Programming 279
19. D/A Converter 280
20. CRC Calculator 282
21. X-Y Conversion 285
21.1 Data Conversion on Reading 286
21.2 Data Conversion on Writing 288
22. Intelligent I/O 289
22.1 Base Timer (for Groups 0 to 2) 304
22.2 Time Measurement (for Groups 0 and 1) 310
22.3 Waveform Generation (for Groups 0 to 2) 314
22.3.1 Single-phase Waveform Output Mode (for Groups 0 to 2) 315
22.3.2 Inverted Waveform Output Mode (for Groups 0 to 2) 317
22.3.3 Set/Reset Waveform Output Mode (SR Waveform Output Mode) (for Groups 0 to 2) 319
22.3.4 Bit Modulation PWM Output Mode (for Group 2) 322
22.3.5 Real-time Port Output Mode (RTP Output Mode) (for Group 2) 324
22.3.6 Parallel Real-time Port Output Mode (RTP Output Mode) (for Group 2) 326
22.4 Group 2 Serial Interface 328
22.4.1 Variable Synchronous Serial Interface Mode (for Group 2) 333
23. Multi-master I2C-bus Interface 336
23.1 Multi-master I2C-bus Interface-associated Registers 338
23.1.1 I2C-bus Transmit/Receive Shift Register (I2CTRSR) 338
23.1.2 I2C-bus Slave Address Register (I2CSAR) 339
23.1.3 I2C-bus Control Register 0 (I2CCR0) 340
23.1.4 I2C-bus Clock Control Register (I2CCCR) 342
23.1.5 I2C-bus START and STOP Conditions Control Register (I2CSSCR) 344
23.1.6 I2C-bus Control Register 1 (I2CCR1) 345
23.1.7 I2C-bus Control Register 2 (I2CCR2) 348
23.1.8 I2C-bus Status Register (I2CSR) 350
23.1.9 I2C-bus Mode Register (I2CMR) 354

23.2 Generating a START Condition 355
23.3 Generating a STOP Condition 357
23.4 START Condition Redundancy Prevention Function 358
23.5 Detecting START and STOP Conditions 359
23.6 Data Transmission and Reception 361
23.6.1 Master Transmission 362
23.6.2 Slave Reception 363
23.7 Notes on Using Multi-master I2C-bus Interface 364
23.7.1 Accessing Multi-master I2C-bus Interface-associated Registers 364
23.7.2 Generating a Repeated START condition 366

24. CAN Module 367

24.1 CAN SFRs 370
24.1.1 CAN0 Control Register (C0CTLR Register) 371
24.1.2 CAN0 Clock Select Register (C0CLKR Register) 375
24.1.3 CAN0 Bit Configuration Register (C0BCR Register) 376
24.1.4 CAN0 Mask Register k (C0MKRk Register) (k = 0 to 7) 378
24.1.5 CAN0 FIFO Received ID Compare Register n (Registers C0FIDCR0 and C0FIDCR1) (n = 0, 1) 379
24.1.6 CAN0 Mask Invalid Register (C0MKIVLR Register) 381
24.1.7 CAN0 Mailbox (C0MBj Register) (j = 0 to 31) 382
24.1.8 CAN0 Mailbox Interrupt Enable Register (C0MIER Register) 386
24.1.9 CAN0 Message Control Register j (C0MCTLj Register) (j = 0 to 31) 387
24.1.10 CAN0 Receive FIFO Control Register (C0RFCR Register) 390
24.1.11 CAN0 Receive FIFO Pointer Control Register (C0RFPCR Register) 393
24.1.12 CAN0 Transmit FIFO Control Register (C0TFCR Register) 394
24.1.13 CAN0 Transmit FIFO Pointer Control Register (C0TFPCR Register) 396
24.1.14 CAN0 Status Register (C0STR Register) 397
24.1.15 CAN0 Mailbox Search Mode Register (C0MSMR Register) 400
24.1.16 CAN0 Mailbox Search Status Register (C0MSSR Register) 401
24.1.17 CAN0 Channel Search Support Register (C0CSSR Register) 403
24.1.18 CAN0 Acceptance Filter Support Register (C0AFSR Register) 404
24.1.19 CAN0 Error Interrupt Enable Register (C0EIER Register) 405
24.1.20 CAN0 Error Interrupt Factor Judge Register (C0EIFR Register) 407
24.1.21 CAN0 Receive Error Count Register (C0RECR Register) 410
24.1.22 CAN0 Transmit Error Count Register (C0TECR Register) 411
24.1.23 CAN0 Error Code Store Register (C0ECSR Register) 412
24.1.24 CAN0 Time Stamp Register (C0TSR Register) 414
24.1.25 CAN0 Test Control Register (C0TCR Register) 415
24.2 Operating Mode 418
24.2.1 CAN Reset Mode 419
24.2.2 CAN Halt Mode 420

24.2.3 CAN Sleep Mode ... 421
24.2.4 CAN Operation Mode (Excluding Bus-off State) ... 422
24.2.5 CAN Operation Mode (Bus-off State) ... 423
24.3 CAN Communication Speed Configuration ... 424
24.3.1 CAN Clock Configuration ... 424
24.3.2 Bit Timing Configuration ... 424
24.3.3 Bit rate ... 425
24.4 Mailbox and Mask Register Structure ... 426
24.5 Acceptance Filtering and Masking Function ... 428
24.6 Reception and Transmission ... 431
24.6.1 Reception ... 432
24.6.2 Transmission ... 434
24.7 CAN Interrupt ... 435

25. I/O Pins 436
25.1 Port Pi Direction Register (PDi Register, i = 0 to 15) ... 437
25.2 Output Function Select Register ... 438
25.3 Input Function Select Register ... 456
25.4 Pull-up Control Registers 0 to 4 (Registers PUR0 to PUR4) ... 460
25.5 Port Control Register (PCR Register) ... 463
25.6 How To Configure Unused Pins ... 464

26. Flash Memory 466
26.1 Overview ... 466
26.2 Flash Memory Protection ... 468
26.2.1 Lock Bit Protection ... 468
26.2.2 ROM Code Protection ... 468
26.2.3 ID Code Protection ... 469
26.3 CPU Rewrite Mode ... 471
26.3.1 Flash Memory Rewrite Bus Timing ... 478
26.3.2 Software Commands ... 482
26.3.3 Mode Transition ... 483
26.3.4 How to Issue Software Commands ... 484
26.3.5 Status Check ... 490
26.4 Standard Serial I/O Mode ... 491
26.5 Parallel I/O mode ... 495
26.6 Notes on Flash Memory Rewriting ... 496
26.6.1 Note on Power Supply ... 496
26.6.2 Note on Hardware Reset ... 496
26.6.3 Note on Flash Memory Protection ... 496
26.6.4 Notes on Programming ... 496
26.6.5 Notes on Interrupts ... 496

26.6.6 Notes on Rewrite Control Program 497
26.6.7 Notes on Number of Programming/Erasure and Software Command Execution Time 497
26.6.8 Other Notes 497

27. Digital Audio Processor (DAP) 498
27.1 A/D Converter 500
27.2 D/A Converter 502
27.3 DAP Control 504
27.3.1 Reset 504
27.3.2 Control Protocol 504
27.4 DAP Control Registers 505
27.4.1 CODEC Control Register (CDCR Register) 506
27.4.2 Audio Interface Configuration Register (AIFC Register) 507
27.4.3 DAP PLL Configuration Register (DPLC Register) 508
27.4.4 DSP Control Register (DSPC Register) 509
27.4.5 DAP Voltage Regulator Control Register (DVRC Register) 510
27.4.6 CODEC Data Selector Control Register (CDSC Register) 511
27.4.7 CODEC A/D Converter Dither Control Register (CADDS Register) 511
27.4.8 CODEC D/A Converter Dither Control Register (CDADS Register) 512
27.4.9 DAP I/O Pin Control Register 0 (DPC0 Register) 512
27.4.10 DAP I/O Pin Control Register 1 (DPC1 Register) 513
27.4.11 DAP I/O Pin Control Register 2 (DPC2 Register) 513
27.4.12 DAP I/O Pin Control Register 3 (DPC3 Register) 514
27.4.13 DAP I/O Pin Control Register 4 (DPC4 Register) 514
27.4.14 DAP I/O Pin Control Register 5 (DPC5 Register) 515
27.4.15 DAP I/O Pin Control Register 6 (DPC6 Register) 516
27.4.16 DAP I/O Pin Control Register 7 (DPC7 Register) 517
27.4.17 DAP I/O Pin Control Register 8 (DPC8 Register) 517
27.4.18 DAP I/O Pin Control Register 10 (DPC10 Register) 518
27.4.19 DAP Input Signal Control Register 0 (DIC0 Register) 518
27.4.20 DAP Input Signal Control Register 1 (DIC1 Register) 519
27.4.21 DAP Protect Register (DPRR Register) 519
27.4.22 DAP Clock Control Register (DCKC Register) 520
27.5 Operation Sequence 521
27.5.1 Start-up Sequence 523
27.5.2 System Clock Configuration 524
27.5.3 Power-off Sequence 525
27.5.4 Power Saving 526

28. Electrical Characteristics 527

29. Usage Notes 557

29.1 Notes on Board Designing ... 557
29.1.1 Power Supply Pins ... 557
29.1.2 Supply Voltage ... 557
29.2 Notes on Register Setting ... 558
29.2.1 Registers with Write-only Bits ... 558
29.3 Notes on Clock Generator ... 560
29.3.1 Sub Clock ... 560
29.3.2 Power Control ... 560
29.4 Notes on Interrupts ... 561
29.4.1 ISP Setting ... 561
29.4.2 NMI ... 561
29.4.3 External Interrupt ... 561
29.5 Notes on DMAC ... 562
29.5.1 DMAC-associated Register Settings ... 562
29.5.2 Read from DMAC-associated Registers ... 562
29.6 Notes on Timers ... 563
29.6.1 Timer A and Timer B ... 563
29.6.2 Timer A ... 563
29.6.3 Timer B ... 565
29.7 Notes on Three-phase Motor Control Timers ... 566
29.7.1 Shutdown ... 566
29.7.2 Register setting ... 566
29.8 Notes on Serial Interface ... 567
29.8.1 Changing the UiBRG Register (i = 0 to 8) ... 567
29.8.2 Synchronous Serial Interface Mode ... 567
29.8.3 Special Mode 1 (I^2C Mode) ... 567
29.8.4 Reset Procedure on Communication Error ... 568
29.9 Notes on A/D Converter ... 569
29.9.1 Notes on Designing Boards ... 569
29.9.2 Notes on Programming ... 570
29.10 Notes on Flash Memory Rewriting ... 571
29.10.1 Note on Power Supply ... 571
29.10.2 Note on Hardware Reset ... 571
29.10.3 Note on Flash Memory Protection ... 571
29.10.4 Notes on Programming ... 571
29.10.5 Notes on Interrupts ... 571
29.10.6 Notes on Rewrite Control Program ... 572
29.10.7 Notes on Number of Programming/Erasure and Software Command Execution Time ... 572

29.10.8 Other Notes 572

Appendix 1. Package Dimensions 573

INDEX 574

1. Overview

1.1 Features

The M16C Family offers a robust platform of 32-/16-bit CISC microcomputers (MCUs) featuring high ROM code efficiency, extensive EMI/EMS noise immunity, ultra-low power consumption, high-speed processing in actual applications, and numerous and varied integrated peripherals. Extensive device scalability from low- to high-end, featuring a single architecture as well as compatible pin assignments and peripheral functions, provides support for a vast range of application fields.

The R32C/100 Series is a high-end microcontroller series in the M16C Family. With a 4-Gbyte memory space, it achieves maximum code efficiency and high-speed processing with 32-bit CISC architecture, multiplier, multiply-accumulate unit, and floating point unit. The selection from the broadest choice of on-chip peripheral devices — UART, CRC, DMAC, A/D and D/A converters, timers, I^2C, and watchdog timer enables to minimize external components.

The R32C/100 Series, in particular, provides the R32C/102 Group as a product specific to audio applications. The embedded audio DSP enables sound field processing operation independently of CPU operation. Since the audio CODEC with two input channels and six output channels is also embedded, sound field processing is available on a single chip. This product, provided as a 176-pin plastic molded LQFP package, configures five channels of audio interface, nine channels of serial interface, one channel of multi-master I^2C-bus interface, and one channel of CAN module.

1.1.1 Applications

Car audio, audio, general industrial equipment, etc.

1.1.2 Performance Overview

Table 1.1 and Table 1.2 list the performance overview of the R32C/102 Group.

Table 1.1 Performance Overview (1/2)

Unit	Function	Explanation
CPU	Central processing unit	R32C/100 Series CPU Core • Basic instructions: 108 • Minimum instruction execution time: 20 ns (f(CPU) = 50 MHz) • Multiplier: 32-bit × 32-bit → 64-bit • Multiply-accumulate unit: 32-bit × 32-bit + 64-bit → 64-bit • IEEE-754 floating point standard: Single precision • 32-bit barrel shifter • Operating mode: Single-chip mode, memory expansion mode, microprocessor mode (optional (1))
Memory		Flash memory: 512 Kbytes/1 Mbyte RAM: 40 K/63 Kbytes Data flash: 4 Kbytes × 2 blocks Refer to Table 1.3 for each product's memory size
Clock	Clock generator	• 4 circuits (main clock, sub clock, PLL, on-chip oscillator) • Oscillation stop detector: Main clock oscillator stop/restart detection • Frequency divide circuit: Divide-by-2 to divide-by-24 selectable • Low power modes: Wait mode, stop mode
Interrupts		Interrupt vectors: 261 External interrupt inputs: $\overline{\text{NMI}}$, $\overline{\text{INT}}$ × 9, key input × 4 Interrupt priority levels: 7
Watchdog Timer		15 bits × 1 (selectable input frequency from prescaler output)
DMA	DMAC	4 channels • Cycle-steal transfer mode • Request sources: 57 • 2 transfer modes: Single transfer, repeat transfer
	DMAC II	• Can be activated by any peripheral interrupt source • 3 transfer functions: Immediate data transfer, calculation transfer, chained transfer
I/O Ports	Programmable I/O ports	• 2 input-only ports • 113 CMOS I/O ports (of which 32 are 5 V tolerant) • A pull-up resistor is selectable for every 4 input ports (except 5 V tolerant inputs)
Timer	Timer A	16-bit timer × 5 Timer mode, event counter mode, one-shot timer mode, pulse-width modulation (PWM) mode Two-phase pulse signal processing in event counter mode (two-phase encoder input) × 3
	Timer B	16-bit timer × 6 Timer mode, event counter mode, pulse frequency measurement mode, pulse-width measurement mode
	Three-phase motor control timer	Three-phase motor control timer × 1 (timers A1, A2, A4, and B2 used) 8-bit programmable dead time timer

Note:
1. Contact a Renesas Electronics sales office to use the optional features.

Table 1.2 Performance Overview (2/2)

Unit	Function	Explanation
Serial Interface	UART0 to UART8 (1)	Asynchronous/synchronous serial interface × 9 channels (1) • I²C-bus (UART0 to UART6) • Special mode 2 (UART0 to UART6) • IEBus (optional (2)) (UART0 to UART6)
A/D Converter		10-bit resolution × 30 channels Sample and hold functionality integrated
D/A Converter		8-bit resolution × 2
CRC Calculator		CRC-CCITT ($X^{16} + X^{12} + X^5 + 1$)
X-Y Converter		16 bits × 16 bits
Intelligent I/O		Time measurement (input capture): 16 bits × 16 Waveform generation (output compare): 16 bits × 24 Serial interface: Variable-length synchronous serial I/O mode, IEBus mode (optional (2))
Multi-master I²C-bus Interface		1 channel
CAN Module		1 channel CAN functionality compliant with ISO11898-1 32 mailboxes
Flash Memory		Programming and erasure supply voltage: VCC = 3.0 to 3.6 V Minimum endurance: 1,000 program/erase cycles Security protection: ROM code protect, ID code protect Debugging: On-chip debug, on-board flash programming
Digital Audio Processor (DAP)	Audio DSP	• 24-bit DualMAC DSP (up to 73.728 MHz) • Data RAM (including delay RAM) • Serial audio interface (SAI): 2 inputs and 3 outputs • Master (synchronous clock output)/slave selectable • DSP performance: Up to 1536 step/fs
	A/D Converter	24-bit resolution × 2 channels
	D/A Converter	24-bit resolution × 6 channels
Operating Frequency/ Supply Voltage	MCU	50 MHz / VCC = 3.0 to 3.6 V
	DSP	73.728 MHz / VCC = 3.0 to 3.6 V
	CODEC	18.432 MHz / CVCC = LVCC = 4.5 to 5.5 V
Operating Temperature		-40°C to 85°C (P version)
Current Consumption	VCC	70 mA (VCC = 3.3 V, f(CPU) = 50 MHz, f(DSP) = 73.728 MHz) 13 µA (VCC = 3.3 V, f(XCIN) = 32.768 kHz, CPU: in wait mode, DAP: stopped)
	CVCC	32 mA (CVCC = LVCC = 5.0 V, VCC = 3.3, CODEC: active) 0.1 µA (CVCC = LVCC = 5.0 V, VCC = 3.3, CODEC: power-off)
	LVCC	450 µA (CVCC = LVCC = 5.0 V, VCC = 3.3, CODEC: active) 0.05 µA (CVCC = LVCC = 5.0 V, VCC = 3.3, CODEC: power-off)
Package		176-pin plastic molded LQFP (PLQP0176KB-A)

Notes:
1. UART8 is the exclusive communication line with the embedded DSP.
2. Contact a Renesas Electronics sales office to use the optional features.

1.2 Product Information

Table 1.3 lists the product information and Figure 1.1 shows the details of the part number.

Table 1.3 R32C/102 Group Product List **As of November, 2010**

Part Number	Package Code (1)	ROM Capacity (2)	RAM Capacity	Remarks
R5J64026PFE	PLQP0176KB-A	512 Kbytes + 8 Kbytes	40 Kbytes	-40°C to 85°C (P version) DSP performance: 1536 step/fs
R5J64026LPFE				-40°C to 85°C (P version) DSP performance:1024 step/fs
R5J64029PFE (D)		1 Mbyte + 8 Kbytes	63 Kbytes	-40°C to 85°C (P version) DSP performance:1536 step/fs

(D): Under development

Notes:
1. The old package code is as follows: PLQP0176KB-A: 176P6Q-A
2. Data flash memory provides an additional 8 Kbytes of ROM.

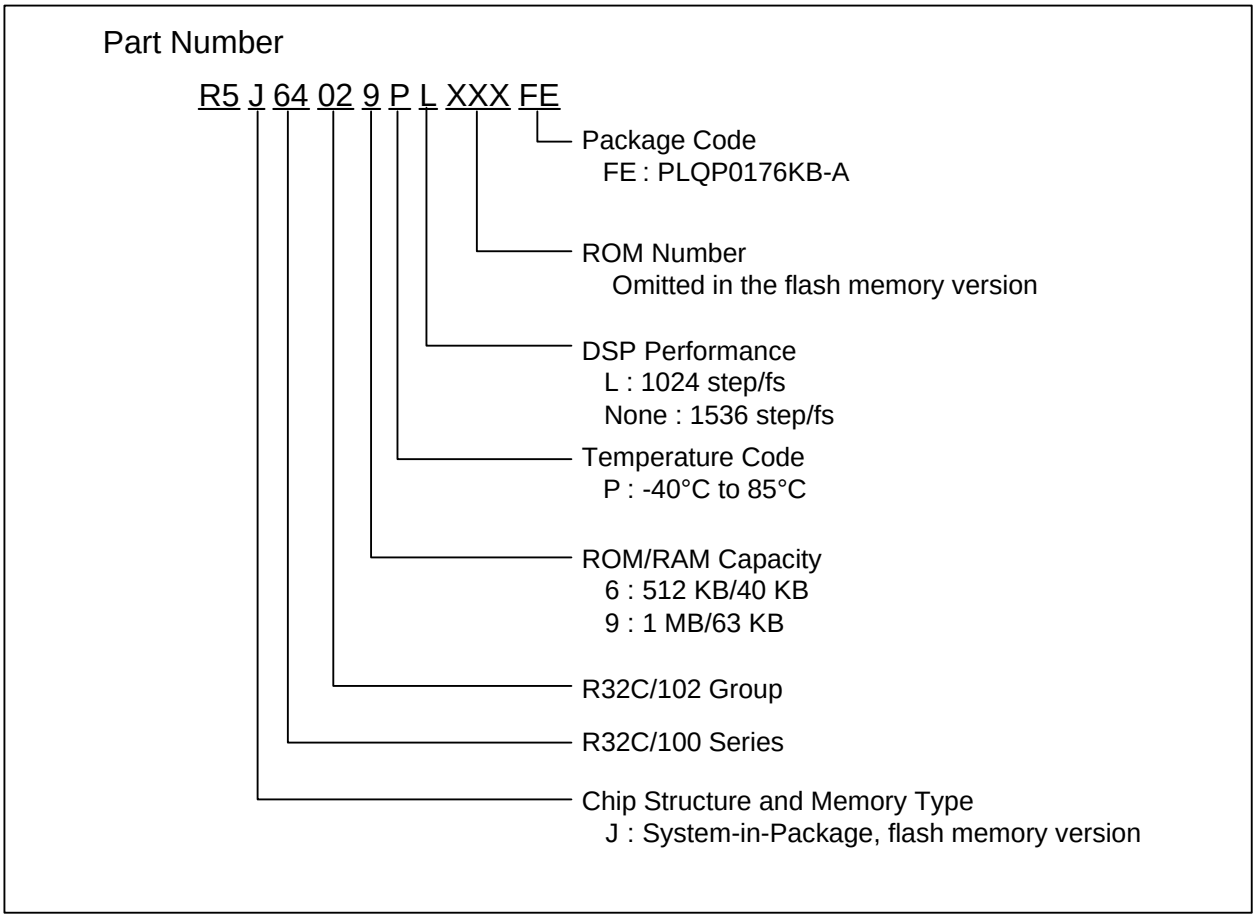


Figure 1.1 Part Numbering

1.3 Block Diagram

Figure 1.2 shows the block diagram for the R32C/102 Group.

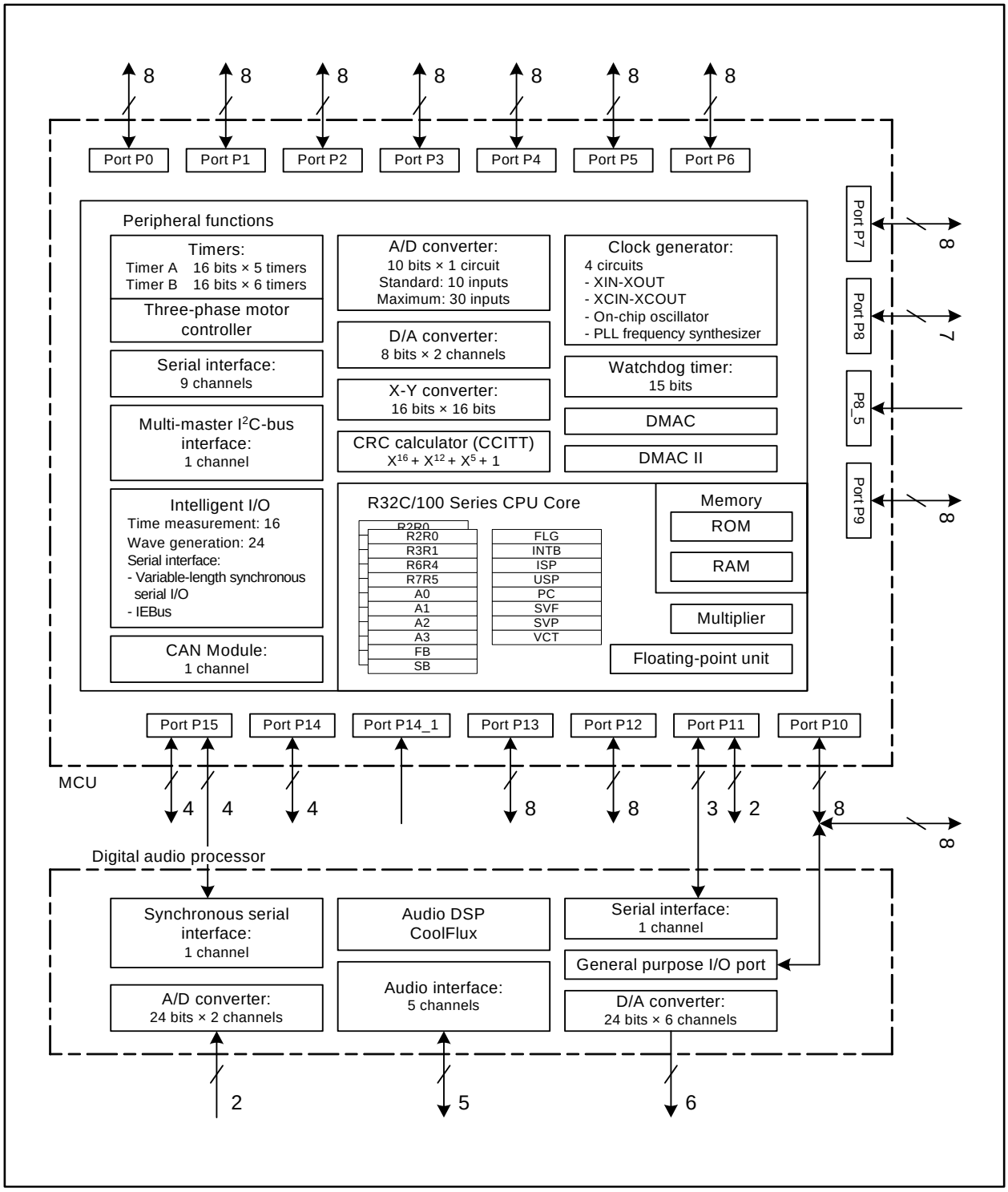
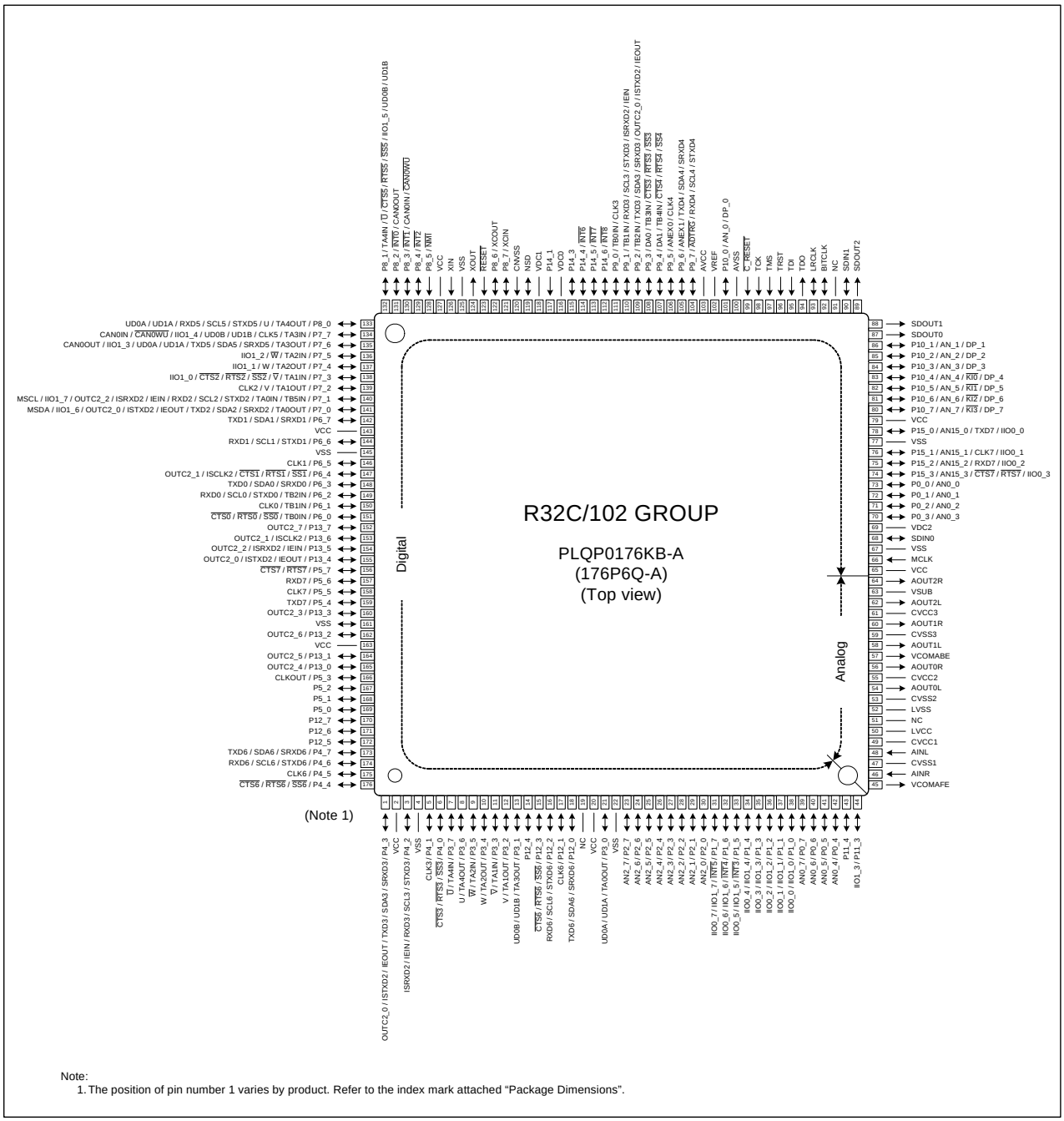


Figure 1.2 R32C/102 Group Block Diagram

1.4 Pin Assignments

Figure 1.3 shows the pin assignments (top view) and Table 1.4 to Table 1.7 list the pin characteristics.



Note:
1. The position of pin number 1 varies by product. Refer to the index mark attached "Package Dimensions".

Figure 1.3 Pin Assignment (top view)

Table 1.4 Pin Characteristics (1/5)

Pin No.	Control Pin	Port	Interrupt Pin	Timer Pin	UART/CAN /Audio I/F Pin	Intelligent I/O Pin	Analog Pin
1		P4_3			TXD3/SDA3/SRXD3	OUTC2_0/ISTXD2/IEOUT	
2	VCC						
3		P4_2			RXD3/SCL3/STXD3	ISRXD2/IEIN	
4	VSS						
5		P4_1			CLK3		
6		P4_0			$\overline{\text{CTS3}}/\overline{\text{RTS3}}/\overline{\text{SS3}}$		
7		P3_7		TA4IN/$\overline{\text{U}}$			
8		P3_6		TA4OUT/U			
9		P3_5		TA2IN/$\overline{\text{W}}$			
10		P3_4		TA2OUT/W			
11		P3_3		TA1IN/$\overline{\text{V}}$			
12		P3_2		TA1OUT/V			
13		P3_1		TA3OUT		UD0B/UD1B	
14		P12_4					
15		P12_3			$\overline{\text{CTS6}}/\overline{\text{RTS6}}/\overline{\text{SS6}}$		
16		P12_2			RXD6/SCL6/STXD6		
17		P12_1			CLK6		
18		P12_0			TXD6/SDA6/SRXD6		
19	NC						
20	VCC						
21		P3_0		TA0OUT		UD0A/UD1A	
22	VSS						
23		P2_7					AN2_7
24		P2_6					AN2_6
25		P2_5					AN2_5
26		P2_4					AN2_4
27		P2_3					AN2_3
28		P2_2					AN2_2
29		P2_1					AN2_1
30		P2_0					AN2_0
31		P1_7	$\overline{\text{INT5}}$			IIO0_7/IIO1_7	
32		P1_6	$\overline{\text{INT4}}$			IIO0_6/IIO1_6	
33		P1_5	$\overline{\text{INT3}}$			IIO0_5/IIO1_5	
34		P1_4				IIO0_4/IIO1_4	
35		P1_3				IIO0_3/IIO1_3	
36		P1_2				IIO0_2/IIO1_2	
37		P1_1				IIO0_1/IIO1_1	
38		P1_0				IIO0_0/IIO1_0	
39		P0_7					AN0_7
40		P0_6					AN0_6
41		P0_5					AN0_5
42		P0_4					AN0_4

Table 1.5 Pin Characteristics (2/5)

Pin No.	Control Pin	Port	Interrupt Pin	Timer Pin	UART/CAN /Audio I/F Pin	Intelligent I/O Pin	Analog Pin
43		P11_4					
44		P11_3				IIO1_3	
45	VCOMAFE						
46							AINR
47	CVSS1						
48							AINL
49	CVCC1						
50	LVCC						
51	NC						
52	LVSS						
53	CVSS2						
54							AOUT0L
55	CVCC2						
56							AOUT0R
57	VCOMABE						
58							AOUT1L
59	CVSS3						
60							AOUT1R
61	CVCC3						
62							AOUT2L
63	VSUB						
64							AOUT2R
65	VCC						
66					MCLK		
67	VSS						
68					SDIN0		
69	VDC2						
70		P0_3					AN0_3
71		P0_2					AN0_2
72		P0_1					AN0_1
73		P0_0					AN0_0
74		P15_3			$\overline{\text{CTS7}}$/$\overline{\text{RTS7}}$	IIO0_3	AN15_3
75		P15_2			RXD7	IIO0_2	AN15_2
76		P15_1			CLK7	IIO0_1	AN15_1
77	VSS						
78		P15_0			TXD7	IIO0_0	AN15_0
79	VCC						
80		P10_7/DP_7	$\overline{\text{KI3}}$				AN_7
81		P10_6/DP_6	$\overline{\text{KI2}}$				AN_6
82		P10_5/DP_5	$\overline{\text{KI1}}$				AN_5
83		P10_4/DP_4	$\overline{\text{KI0}}$				AN_4
84		P10_3/DP_3					AN_3

Table 1.6 Pin Characteristics (3/5)

Pin No.	Control Pin	Port	Interrupt Pin	Timer Pin	UART/CAN /Audio I/F Pin	Intelligent I/O Pin	Analog Pin
85		P10_2/DP_2					AN_2
86		P10_1/DP_1					AN_1
87					SDOUT0		
88					SDOUT1		
89					SDOUT2		
90					SDIN1		
91	NC						
92					BITCLK		
93					LRCLK		
94	TDO						
95	TDI						
96	TRST						
97	TMS						
98	TCK						
99	$\overline{\text{C\_RESET}}$						
100	AVSS						
101		P10_0/DP_0					AN_0
102	VREF						
103	AVCC						
104		P9_7			RXD4/SCL4/STXD4		$\overline{\text{ADTRG}}$
105		P9_6			TXD4/SDA4/SRXD4		ANEX1
106		P9_5			CLK4		ANEX0
107		P9_4		TB4IN	$\overline{\text{CTS4}}$/$\overline{\text{RTS4}}$/$\overline{\text{SS4}}$		DA1
108		P9_3		TB3IN	$\overline{\text{CTS3}}$/$\overline{\text{RTS3}}$/$\overline{\text{SS3}}$		DA0
109		P9_2		TB2IN	TXD3/SDA3/SRXD3	OUTC2_0/ISTXD2/ IEOUT	
110		P9_1		TB1IN	RXD3/SCL3/STXD3	ISRXD2/IEIN	
111		P9_0		TB0IN	CLK3		
112		P14_6	$\overline{\text{INT8}}$				
113		P14_5	$\overline{\text{INT7}}$				
114		P14_4	$\overline{\text{INT6}}$				
115		P14_3					
116	VDC0						
117		P14_1					
118	VDC1						
119	NSD						
120	CNVSS						
121	XCIN	P8_7					
122	XCOUT	P8_6					
123	$\overline{\text{RESET}}$						
124	XOUT						
125	VSS						
126	XIN						

Table 1.7 Pin Characteristics (4/5)

Pin No.	Control Pin	Port	Interrupt Pin	Timer Pin	UART/CAN /Audio I/F Pin	Intelligent I/O Pin	Analog Pin
127	VCC						
128		P8_5	$\overline{NMI}$				
129		P8_4	$\overline{INT2}$				
130		P8_3	$\overline{INT1}$		CAN0IN/$\overline{CAN0WU}$		
131		P8_2	$\overline{INT0}$		CAN0OUT		
132		P8_1		TA4IN/$\overline{U}$	$\overline{CTS5}$/$\overline{RTS5}$/$\overline{SS5}$	IIO1_5/UD0B/UD1B	
133		P8_0		TA4OUT/U	RXD5/SCL5/STXD5	UD0A/UD1A	
134		P7_7		TA3IN	CLK5/CAN0IN/ $\overline{CAN0WU}$	IIO1_4/UD0B/UD1B	
135		P7_6		TA3OUT	TXD5/SDA5/SRXD5/ CAN0OUT	IIO1_3/UD0A/UD1A	
136		P7_5		TA2IN/$\overline{W}$		IIO1_2	
137		P7_4		TA2OUT/W		IIO1_1	
138		P7_3		TA1IN/$\overline{V}$	$\overline{CTS2}$/$\overline{RTS2}$/$\overline{SS2}$	IIO1_0	
139		P7_2		TA1OUT/V	CLK2		
140		P7_1		TA0IN/ TB5IN	RXD2/SCL2/STXD2/ MSCL	IIO1_7/OUTC2_2/ISRXD2/IEIN	
141		P7_0		TA0OUT	TXD2/SDA2/SRXD2/ MSDA	IIO1_6/OUTC2_0/ISTXD2/ IEOUT	
142		P6_7			TXD1/SDA1/SRXD1		
143	VCC						
144		P6_6			RXD1/SCL1/STXD1		
145	VSS						
146		P6_5			CLK1		
147		P6_4			$\overline{CTS1}$/$\overline{RTS1}$/$\overline{SS1}$	OUTC2_1/ISCLK2	
148		P6_3			TXD0/SDA0/SRXD0		
149		P6_2		TB2IN	RXD0/SCL0/STXD0		
150		P6_1		TB1IN	CLK0		
151		P6_0		TB0IN	$\overline{CTS0}$/$\overline{RTS0}$/$\overline{SS0}$		
152		P13_7				OUTC2_7	
153		P13_6				OUTC2_1/ISCLK2	
154		P13_5				OUTC2_2/ISRXD2/IEIN	
155		P13_4				OUTC2_0/ISTXD2/IEOUT	
156		P5_7			$\overline{CTS7}$/$\overline{RTS7}$		
157		P5_6			RXD7		
158		P5_5			CLK7		
159		P5_4			TXD7		
160		P13_3				OUTC2_3	
161	VSS						
162		P13_2				OUTC2_6	
163	VCC						
164		P13_1				OUTC2_5	
165		P13_0				OUTC2_4	
166	CLKOUT	P5_3					

Table 1.8 Pin Characteristics (5/5)

Pin No.	Control Pin	Port	Interrupt Pin	Timer Pin	UART/CAN /Audio I/F Pin	Intelligent I/O Pin	Analog Pin
167		P5_2					
168		P5_1					
169		P5_0					
170		P12_7					
171		P12_6					
172		P12_5					
173		P4_7			TXD6/SDA6/SRXD6		
174		P4_6			RXD6/SCL6/STXD6		
175		P4_5			CLK6		
176		P4_4			$\overline{\text{CTS6}}$/$\overline{\text{RTS6}}$/$\overline{\text{SS6}}$		

1.5 Pin Definitions and Functions

Table 1.9 to Table 1.11 list the pin definitions and functions.

Table 1.9 Pin Definitions and Functions (1/3)

Function	Symbol	I/O	Description
Power supply	VCC, VSS	I	Applicable as follows: VCC = 3.0 to 3.6 V, VSS = 0 V
	LVCC, LVSS	I	Power supply for the level shifter of CODEC. Applicable as follows: LVCC = 4.5 to 5.5 V, LVSS = 0 V
	VSUB	I	This pin is to stabilize the electrical potential on the semiconductor substrate. VSUB should be 0 V
Connecting pins for decoupling capacitor	VDC0, VDC1	—	A decoupling capacitor for internal voltage should be connected between VDC0 and VDC1
	VDC2	—	A decoupling capacitor for internal voltage should be connected between VDC2 and VSS
Analog power supply	AVCC, AVSS	I	Power supply for the A/D converter. AVCC and AVSS should be connected to VCC and VSS, respectively
	CVCC1 to CVCC3 CVSS1 to CVSS3	I	Power supply for the analog circuit of CODEC. Applicable as follows: CVCC1 to CVCC3 = 4.5 to 5.5 V, CVSS1 to CVSS3 = 0 V
	VCOMAFE, VCOMABE	O	Midpoint potential output pin for CODEC. A ceramic capacitor (0.1 µF approx.) and an electrolytic capacitor (100 µF approx.) should be connected in parallel between VCOMAFE and CVSS1, and between VCOMABE and CVSS2
Reset input	$\overline{\text{RESET}}$	I	The MCU is reset when this pin is driven low
	$\overline{\text{C\_RESET}}$	I	The digital audio processor is reset when this pin is driven low
CNVSS	CNVSS	I	This pin should be connected to VSS via a resistor
Debug port	NSD	I/O	This pin is to communicate with a debugger. It should be connected to VCC via a resistor of 1 to 4.7 kΩ
Main clock input	XIN	I	Input/output for the main clock oscillator. A crystal, or a ceramic resonator should be connected between pins XIN and XOUT. An external clock should be input at the XIN while leaving the XOUT open
Main clock output	XOUT	O	
Sub clock input	XCIN	I	Input/output for the sub clock oscillator. A crystal oscillator should be connected between pins XCIN and XCOUT. An external clock should be input at the XCIN while leaving the XCOUT open
Sub clock output	XCOUT	O	
Clock output	CLKOUT	O	Output of the clock with the same frequency as low speed clocks, f8, or f32
External interrupt input	$\overline{\text{INT0}}$ to $\overline{\text{INT8}}$	I	Input for external interrupts
NMI input	P8_5/$\overline{\text{NMI}}$	I	Input for NMI
Key input interrupt	$\overline{\text{KI0}}$ to $\overline{\text{KI3}}$	I	Input for the key input interrupt

Table 1.10 Pin Definitions and Functions (2/3)

Function	Symbol	I/O	Description
I/O port	P0_0 to P0_7, P1_0 to P1_7, P2_0 to P2_7, P3_0 to P3_7, P4_0 to P4_7, P5_0 to P5_7, P6_0 to P6_7, P7_0 to P7_7, P8_0 to P8_4, P8_6, P8_7, P9_0 to P9_7, P10_0 to P10_7, P11_3, P11_4, P12_0 to P12_7, P13_0 to P13_7, P14_3 to P14_6, P15_0 to P15_3	I/O	I/O ports in CMOS. Each port can be programmed to input or output under the control of the direction register. Some ports are 5 V tolerant inputs. Pull-up resistors and N-channel open drain setting can be enabled on some ports. Refer to Table 1.12 “Pin Specifications” for details
	DP_0 to DP_7	I/O	I/O ports in CMOS. Each port can be programmed to input or output under the control of the direction register. These ports are connected to P10_0 to P10_7 in a wired-OR configuration.
Input port	P14_1	I	Input port in CMOS Pull-up resistor is selectable Refer to Table 1.12 “Pin Specifications” for details
Timer A	TA0OUT to TA4OUT	I/O	Timers A0 to A4 input/output
	TA0IN to TA4IN	I	Timers A0 to A4 input
Timer B	TB0IN to TB5IN	I	Timers B0 to B5 input
Three-phase motor control timer output	U, $\overline{U}$, V, $\overline{V}$, W, $\overline{W}$	O	Three-phase motor control timer output
Serial interface	$\overline{CTS0}$ to $\overline{CTS7}$	I	Handshake input
	$\overline{RTS0}$ to $\overline{RTS7}$	O	Handshake output
	CLK0 to CLK7	I/O	Transmit/receive clock input/output
	RXD0 to RXD7	I	Serial data input
	TXD0 to TXD7	O	Serial data output
I^2C-bus (simplified)	SDA0 to SDA6	I/O	Serial data input/output
	SCL0 to SCL6	I/O	Transmit/receive clock input/output
Serial interface special functions	STXD0 to STXD6	O	Serial data output in slave mode
	SRXD0 to SRXD6	I	Serial data input in slave mode
	$\overline{SS0}$ to $\overline{SS6}$	I	Input to control serial interface special functions

Table 1.11 Pin Definitions and Functions (3/3)

Function	Symbol	I/O	Description
A/D converter	AN_0 to AN_7, AN0_0 to AN0_7, AN2_0 to AN2_7, AN15_0 to AN15_3	I	Analog input for the A/D converter
	$\overline{\text{ADTRG}}$	I	External trigger input for the A/D converter
	ANEX0	I/O	Expanded analog input for the A/D converter and output in external op-amp connection mode
	ANEX1	I	Expanded analog input for the A/D converter
D/A converter	DA0, DA1	O	Output for the D/A converter
Reference voltage input	VREF	I	Reference voltage input for the A/D converter and D/A converter
Intelligent I/O	IIO0_0 to IIO0_7	I/O	Input/output for the Intelligent I/O group 0. Either input capture or output compare is selectable
	IIO1_0 to IIO1_7	I/O	Input/output for the Intelligent I/O group 1. Either input capture or output compare is selectable
	UD0A, UD0B, UD1A, UD1B	I	Input for the two-phase encoder
	OUTC2_0 to OUTC2_7	O	Output for OC (output compare) of the Intelligent I/O group 2
	ISCLK2	I/O	Clock input/output for the serial interface
	ISRXD2	I	Receive data input for the serial interface
	ISTXD2	O	Transmit data output for the serial interface
	IEIN	I	Receive data input for the serial interface
	IEOUT	O	Transmit data output for the serial interface
Multi-master I^2C-bus	MSDA	I/O	Serial data input/output
	MSCL	I/O	Transmit/receive clock input/output
CAN Module	CAN0IN	I	Receive data input for the CAN communications
	CAN0OUT	O	Transmit data output for the CAN communications
	$\overline{\text{CAN0WU}}$	I	Input for the CAN wake-up interrupt
Audio Interface	SDIN0, SDIN1	I	Audio data input
	SDOUT0 to SDOUT2	O	Audio data output
	LRCLK	I/O	LR clock input/output
	BITCLK	I/O	Transmit/receive clock input/output
	MCLK	I	Reference clock for the audio interface
Digital Audio Processor	AINL	I	Signal input for left channel to the A/D converter
	AINR	I	Signal input for right channel to the A/D converter
	AOUT0L to AOUT2L	O	Signal output for left channel from the D/A converter
	AOUT0R to AOUT2R	O	Signal output for right channel from the D/A converter

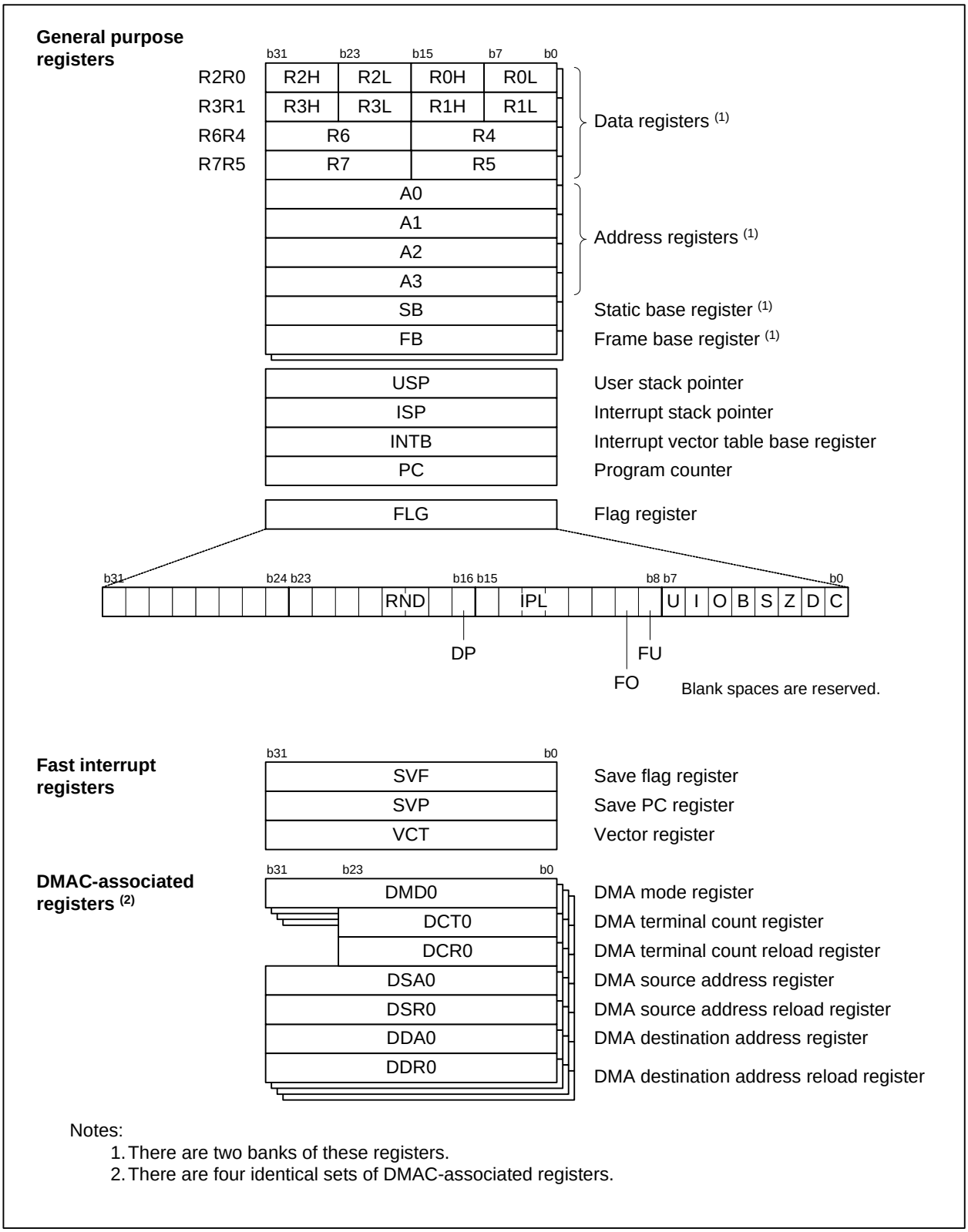
Table 1.12 Pin Specifications

Pin names	Selectable Functions		
	Pull-up resistor (1)	N-channel open drain (2)	5 V tolerant input (3)
P0_0 to P0_7	✓		
P1_0 to P1_7	✓		
P2_0 to P2_7	✓		
P3_0 to P3_7	✓		
P4_0 to P4_7		✓	✓
P5_0 to P5_3	✓		
P5_4 to P5_7		✓	✓
P6_0 to P6_7		✓	✓
P7_0 to P7_7		✓	✓
P8_0 to P8_3		✓	✓
P8_4, P8_6, P8_7	✓		
P9_0 to P9_7	✓	✓	
P10_0 to P10_7	✓		
P11_3	✓	✓	
P11_4	✓		
P12_0 to P12_3	✓	✓	
P12_4 to P12_7	✓		
P13_0 to P13_7	✓		
P14_1, P14_3	✓		
P14_4 to P14_6	✓		
P15_0 to P15_3	✓	✓	

Notes:
1. Pull-up resistors are selected in 4-pin units, but are only enabled for those pins set as input pins.
2. N-channel open drain output can be enabled on the applicable pins on a discrete pin basis.
3. 5 V tolerant input is enabled when an applicable pin is set as an input port. When it is set as an I/O port, to enable 5 V tolerant input, this pin should be set as N-channel open drain output.

2. Central Processing Unit (CPU)

The CPU contains the registers shown below. There are two register banks each consisting of registers R2R0, R3R1, R6R4, R7R5, A0 to A3, SB, and FB.



Notes:
1. There are two banks of these registers.
2. There are four identical sets of DMAC-associated registers.

Figure 2.1 CPU Registers

2.1 General Purpose Registers

2.1.1 Data Registers (R2R0, R3R1, R6R4, and R7R5)

These 32-bit registers are primarily used for transfers and arithmetic/logic operations.
Each of the registers can be divided into upper and lower 16-bit registers, e.g. R2R0 can be divided into R2 and R0, R3R0 can be divided into R3 and R1, etc.
Moreover, data registers R2R0 and R3R1 can be divided into four 8-bit data registers: upper (R2H and R3H), mid-upper (R2L and R3L), mid-lower (R0H and R1H), and lower (R0L and R1L).

2.1.2 Address Registers (A0, A1, A2, and A3)

These 32-bit registers have functions similar to data registers. They are also used for address register indirect addressing and address register relative addressing.

2.1.3 Static Base Register (SB)

This 32-bit register is used for SB relative addressing.

2.1.4 Frame Base Register (FB)

This 32-bit register is used for FB relative addressing.

2.1.5 Program Counter (PC)

This 32-bit counter indicates the address of the instruction to be executed next.

2.1.6 Interrupt Vector Table Base Register (INTB)

This 32-bit register indicates the start address of a relocatable vector table.

2.1.7 User Stack Pointer (USP) and Interrupt Stack Pointer (ISP)

Two types of 32-bit stack pointers (SPs) are provided: user stack pointer (USP) and interrupt stack pointer (ISP).
Use the stack pointer select flag (U flag) to select either the user stack pointer (USP) or the interrupt stack pointer (ISP). The U flag is bit 7 in the flag register (FLG). Refer to 2.1.8 “Flag Register (FLG)” for details.
To minimize the overhead of interrupt sequence due to less memory access, set the user stack pointer (USP) or the interrupt stack pointer (ISP) to a multiple of 4.

2.1.8 Flag Register (FLG)

This 32-bit register indicates the CPU status.

2.1.8.1 Carry Flag (C flag)

This flag retains a carry, borrow, or shifted-out bit generated by the arithmetic logic unit (ALU).

2.1.8.2 Debug Flag (D flag)

This flag is only for debugging. Only set this bit to 0.

2.1.8.3 Zero Flag (Z flag)

This flag becomes 1 when the result of an operation is 0; otherwise it is 0.

2.1.8.4 Sign Flag (S flag)

This flag becomes 1 when the result of an operation is a negative value; otherwise it is 0.

2.1.8.5 Register Bank Select Flag (B flag)

This flag selects a register bank. It indicates 0 when register bank 0 is selected, and 1 when register bank 1 is selected.

2.1.8.6 Overflow Flag (O flag)

This flag becomes 1 when the result of an operation overflows; otherwise it is 0.

2.1.8.7 Interrupt Enable Flag (I flag)

This flag enables maskable interrupts. To disable maskable interrupts, set this flag to 0. To enable them, set this flag to 1. When an interrupt is accepted, the flag becomes 0.

2.1.8.8 Stack Pointer Select Flag (U flag)

To select the interrupt stack pointer (ISP), set this flag to 0. To select the user stack pointer (USP), set this flag to 1.
It becomes 0 when a hardware interrupt is accepted or when an INT instruction designated by a software interrupt number from 0 to 127 is executed.

2.1.8.9 Floating-point Underflow Flag (FU flag)

This flag becomes 1 when an underflow occurs in a floating-point operation; otherwise it is 0. It also becomes 1 when the operand contains invalid numbers (subnormal numbers).

2.1.8.10 Floating-point Overflow Flag (FO flag)

This flag becomes 1 when an overflow occurs in a floating-point operation; otherwise it is 0. It also becomes 1 when the operand contains invalid numbers (subnormal numbers).

2.1.8.11 Processor Interrupt Priority Level (IPL)

The processor interrupt priority level (IPL), consisting of 3 bits, selects a processor interrupt priority level from level 0 to 7. An interrupt is enabled when the interrupt request level is higher than the selected IPL.
When the processor interrupt priority level (IPL) is set to 111b (level 7), all interrupts are disabled.

2.1.8.12 Fixed-point Radix Point Designation Bit (DP bit)

This bit designates the radix point. It also specifies which portion of the fixed-point multiplication result to extract. It is used for the MULX instruction.

2.1.8.13 Floating-point Rounding Mode (RND)

The 2-bit floating-point rounding mode selects a rounding mode for floating-point calculation results.

2.1.8.14 Reserved

Only set this bit to 0. The read value is undefined.

2.2 Fast Interrupt Registers

The following three registers are provided to minimize the overhead of the interrupt sequence. Refer to 10.4 “Fast Interrupt” for details.

2.2.1 Save Flag Register (SVF)

This 32-bit register is used to save the flag register when a fast interrupt is generated.

2.2.2 Save PC Register (SVP)

This 32-bit register is used to save the program counter when a fast interrupt is generated.

2.2.3 Vector Register (VCT)

This 32-bit register is used to indicate a jump address when a fast interrupt is generated.

2.3 DMAC-associated Registers

There are seven types of DMAC-associated registers. Refer to 12. “DMAC” for details.

2.3.1 DMA Mode Registers (DMD0, DMD1, DMD2, and DMD3)

These 32-bit registers are used to set DMA transfer mode, bit rate, etc.

2.3.2 DMA Terminal Count Registers (DCT0, DCT1, DCT2, and DCT3)

These 24-bit registers are used to set the number of DMA transfers.

2.3.3 DMA Terminal Count Reload Registers (DCR0, DCR1, DCR2, and DCR3)

These 24-bit registers are used to set the reloaded values for DMA terminal count registers.

2.3.4 DMA Source Address Registers (DSA0, DSA1, DSA2, and DSA3)

These 32-bit registers are used to set a DMA source address.

2.3.5 DMA Source Address Reload Registers (DSR0, DSR1, DSR2, and DSR3)

These 32-bit registers are used to set the reloaded value for DMA source address register.

2.3.6 DMA Destination Address Registers (DDA0, DDA1, DDA2, and DDA3)

These 32-bit registers are used to set a DMA destination address.

2.3.7 DMA Destination Address Reload Registers (DDR0, DDR1, DDR2, and DDR3)

These 32-bit registers are used to set reloaded values for DMA destination address registers.

3. Memory

Figure 3.1 shows the memory map of the R32C/102 Group.
The R32C/102 Group provides a 4-Gbyte address space from 00000000h to FFFFFFFFh.

The internal ROM is mapped from address FFFFFFFFh in the inferior direction. For example, the 1-Mbyte internal ROM is mapped from FFF00000h to FFFFFFFFh.
The fixed interrupt vector table contains the start address of interrupt handlers and is mapped from FFFFFFDCh to FFFFFFFFh.
The internal RAM is mapped from address 00000400h in the superior direction. For example, the 63-Kbyte internal RAM is mapped from 00000400h to 0000FFFFh. Besides being used for data storage, the internal RAM functions as a stack(s) for subroutine calls and/or interrupt handlers.

Special function registers (SFRs), which are control registers for peripheral functions, are mapped from 00000000h to 000003FFh, and from 00040000h to 0004FFFFh. Unoccupied SFR locations are reserved, and no access is allowed.

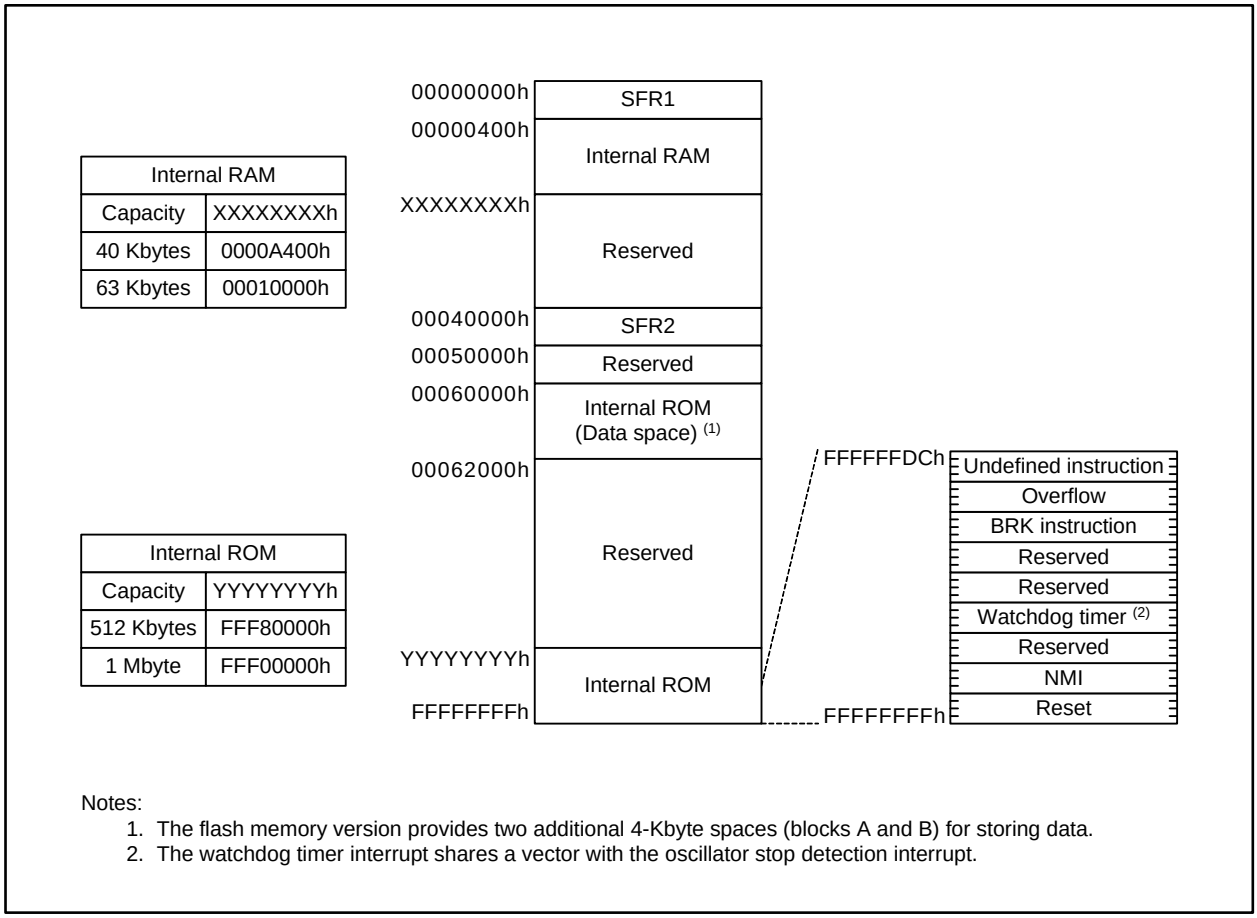


Figure 3.1 Memory Map

4. Special Function Registers (SFRs)

SFRs are memory-mapped peripheral registers that control the operation of peripherals. Table 4.1 SFR List (1) to Table 4.39 SFR List (39) list the SFR details.

Table 4.1 SFR List (1)

Address	Register	Symbol	Reset Value
000000h			
000001h			
000002h			
000003h			
000004h	Clock Control Register	CCR	0001 1000b
000005h			
000006h	Flash Memory Control Register	FMCR	0000 0001b
000007h	Protect Release Register	PRR	00h
000008h			
000009h			
00000Ah			
00000Bh			
00000Ch			
00000Dh			
00000Eh			
00000Fh			
000010h			
000011h			
000012h			
000013h			
000014h			
000015h			
000016h			
000017h			
000018h			
000019h			
00001Ah			
00001Bh			
00001Ch	Flash Memory Rewrite Bus Control Register	FEBC	0000h
00001Dh			
00001Eh	Peripheral Bus Control Register	PBC	0504h
00001Fh			
000020h to 00005Fh			

X: Undefined

Blanks are reserved. No access is allowed.

Table 4.2 SFR List (2)

Address	Register	Symbol	Reset Value
000060h			
000061h	Timer B5 Interrupt Control Register	TB5IC	XXXX X000b
000062h	UART5 Transmit/NACK Interrupt Control Register	S5TIC	XXXX X000b
000063h	UART2 Receive/ACK Interrupt Control Register/I2C-bus Line Interrupt Control Register	S2RIC/I2CLIC	XXXX X000b
000064h	UART6 Transmit/NACK Interrupt Control Register	S6TIC	XXXX X000b
000065h	UART3 Receive/ACK Interrupt Control Register	S3RIC	XXXX X000b
000066h	UART5/6 Bus Collision, Start Condition/Stop Condition Detection Interrupt Control Register	BCN5IC/BCN6IC	XXXX X000b
000067h	UART4 Receive/ACK Interrupt Control Register	S4RIC	XXXX X000b
000068h	DMA0 Transfer Complete Interrupt Control Register	DM0IC	XXXX X000b
000069h	UART0/3 Bus Collision, Start Condition/Stop Condition Detection Interrupt Control Register	BCN0IC/BCN3IC	XXXX X000b
00006Ah	DMA2 Transfer Complete Interrupt Control Register	DM2IC	XXXX X000b
00006Bh	A/D Converter 0 Convert Completion Interrupt Control Register	AD0IC	XXXX X000b
00006Ch	Timer A0 Interrupt Control Register	TA0IC	XXXX X000b
00006Dh	Intelligent I/O Interrupt Control Register 0	IIO0IC	XXXX X000b
00006Eh	Timer A2 Interrupt Control Register	TA2IC	XXXX X000b
00006Fh	Intelligent I/O Interrupt Control Register 2	IIO2IC	XXXX X000b
000070h	Timer A4 Interrupt Control Register	TA4IC	XXXX X000b
000071h	Intelligent I/O Interrupt Control Register 4	IIO4IC	XXXX X000b
000072h	UART0 Receive/ACK Interrupt Control Register	S0RIC	XXXX X000b
000073h	Intelligent I/O Interrupt Control Register 6	IIO6IC	XXXX X000b
000074h	UART1 Receive/ACK Interrupt Control Register	S1RIC	XXXX X000b
000075h	Intelligent I/O Interrupt Control Register 8	IIO8IC	XXXX X000b
000076h	Timer B1 Interrupt Control Register	TB1IC	XXXX X000b
000077h	Intelligent I/O Interrupt Control Register 10	IIO10IC	XXXX X000b
000078h	Timer B3 Interrupt Control Register	TB3IC	XXXX X000b
000079h			
00007Ah	INT5 Interrupt Control Register	INT5IC	XX00 X000b
00007Bh	CAN0 Wake-up Interrupt Control Register	C0WIC	XXXX X000b
00007Ch	INT3 Interrupt Control Register	INT3IC	XX00 X000b
00007Dh			
00007Eh	INT1 Interrupt Control Register	INT1IC	XX00 X000b
00007Fh			
000080h			
000081h	UART2 Transmit/NACK Interrupt Control Register/I2C-bus Interrupt Control Register	S2TIC/I2CIC	XXXX X000b
000082h	UART5 Receive/ACK Interrupt Control Register	S5RIC	XXXX X000b
000083h	UART3 Transmit/NACK Interrupt Control Register	S3TIC	XXXX X000b
000084h	UART6 Receive/ACK Interrupt Control Register	S6RIC	XXXX X000b
000085h	UART4 Transmit/NACK Interrupt Control Register	S4TIC	XXXX X000b
000086h			
000087h	UART2 Bus Collision, Start Condition/Stop Condition Detection Interrupt Control Register	BCN2IC	XXXX X000b

X: Undefined

Blanks are reserved. No access is allowed.

Table 4.3 SFR List (3)

Address	Register	Symbol	Reset Value
000088h	DMA1 Transfer Complete Interrupt Control Register	DM1IC	XXXX X000b
000089h	UART1/4 Bus Collision, Start Condition/Stop Condition Detection Interrupt Control Register	BCN1IC/BCN4IC	XXXX X000b
00008Ah	DMA3 Transfer Complete Interrupt Control Register	DM3IC	XXXX X000b
00008Bh	Key Input Interrupt Control Register	KUPIC	XXXX X000b
00008Ch	Timer A1 Interrupt Control Register	TA1IC	XXXX X000b
00008Dh	Intelligent I/O Interrupt Control Register 1	IIO1IC	XXXX X000b
00008Eh	Timer A3 Interrupt Control Register	TA3IC	XXXX X000b
00008Fh	Intelligent I/O Interrupt Control Register 3	IIO3IC	XXXX X000b
000090h	UART0 Transmit/NACK Interrupt Control Register	S0TIC	XXXX X000b
000091h	Intelligent I/O Interrupt Control Register 5	IIO5IC	XXXX X000b
000092h	UART1 Transmit/NACK Interrupt Control Register	S1TIC	XXXX X000b
000093h	Intelligent I/O Interrupt Control Register 7	IIO7IC	XXXX X000b
000094h	Timer B0 Interrupt Control Register	TB0IC	XXXX X000b
000095h	Intelligent I/O Interrupt Control Register 9	IIO9IC	XXXX X000b
000096h	Timer B2 Interrupt Control Register	TB2IC	XXXX X000b
000097h	Intelligent I/O Interrupt Control Register 11	IIO11IC	XXXX X000b
000098h	Timer B4 Interrupt Control Register	TB4IC	XXXX X000b
000099h			
00009Ah	INT4 Interrupt Control Register	INT4IC	XX00 X000b
00009Bh			
00009Ch	INT2 Interrupt Control Register	INT2IC	XX00 X000b
00009Dh			
00009Eh	INT0 Interrupt Control Register	INT0IC	XX00 X000b
00009Fh			
0000A0h	Intelligent I/O Interrupt Request Register 0	IIO0IR	0000 0XX1b
0000A1h	Intelligent I/O Interrupt Request Register 1	IIO1IR	0000 0XX1b
0000A2h	Intelligent I/O Interrupt Request Register 2	IIO2IR	0000 0X01b
0000A3h	Intelligent I/O Interrupt Request Register 3	IIO3IR	0000 XXX1b
0000A4h	Intelligent I/O Interrupt Request Register 4	IIO4IR	000X 0XX1b
0000A5h	Intelligent I/O Interrupt Request Register 5	IIO5IR	000X 0XX1b
0000A6h	Intelligent I/O Interrupt Request Register 6	IIO6IR	000X 0XX1b
0000A7h	Intelligent I/O Interrupt Request Register 7	IIO7IR	X00X 0XX1b
0000A8h	Intelligent I/O Interrupt Request Register 8	IIO8IR	XX0X 0XX1b
0000A9h	Intelligent I/O Interrupt Request Register 9	IIO9IR	0X00 0XX1b
0000AAh	Intelligent I/O Interrupt Request Register 10	IIO10IR	0X00 0XX1b
0000ABh	Intelligent I/O Interrupt Request Register 11	IIO11IR	0X00 0XX1b
0000ACh			
0000ADh			
0000AEh			
0000AFh			

X: Undefined

Blanks are reserved. No access is allowed.

Table 4.4 SFR List (4)

Address	Register	Symbol	Reset Value
0000B0h	Intelligent I/O Interrupt Enable Register 0	IIO0IE	00h
0000B1h	Intelligent I/O Interrupt Enable Register 1	IIO1IE	00h
0000B2h	Intelligent I/O Interrupt Enable Register 2	IIO2IE	00h
0000B3h	Intelligent I/O Interrupt Enable Register 3	IIO3IE	00h
0000B4h	Intelligent I/O Interrupt Enable Register 4	IIO4IE	00h
0000B5h	Intelligent I/O Interrupt Enable Register 5	IIO5IE	00h
0000B6h	Intelligent I/O Interrupt Enable Register 6	IIO6IE	00h
0000B7h	Intelligent I/O Interrupt Enable Register 7	IIO7IE	00h
0000B8h	Intelligent I/O Interrupt Enable Register 8	IIO8IE	00h
0000B9h	Intelligent I/O Interrupt Enable Register 9	IIO9IE	00h
0000BAh	Intelligent I/O Interrupt Enable Register 10	IIO10IE	00h
0000BBh	Intelligent I/O Interrupt Enable Register 11	IIO11IE	00h
0000BCh			
0000BDh			
0000BEh			
0000BFh			
0000C0h			
0000C1h	CAN0 Transmit Interrupt Control Register	C0TIC	XXXX X000b
0000C2h			
0000C3h	CAN0 Error Interrupt Control Register	C0EIC	XXXX X000b
0000C4h			
0000C5h			
0000C6h			
0000C7h			
0000C8h			
0000C9h			
0000CAh			
0000CBh			
0000CCh			
0000CDh			
0000CEh			
0000CFh			
0000D0h	CAN0 Transmit FIFO Interrupt Control Register	C0FTIC	XXXX X000b
0000D1h			
0000D2h			
0000D3h			
0000D4h			
0000D5h			
0000D6h			
0000D7h			
0000D8h			
0000D9h			
0000DAh			
0000DBh			
0000DCh			
0000DDh	UART7 Transmit Interrupt Control Register	S7TIC	XXXX X000b
0000DEh	INT7 Interrupt Control Register	INT7IC	XX00 X000b
0000DFh	UART8 Transmit Interrupt Control Register	S8TIC	XXXX X000b

X: Undefined
Blanks are reserved. No access is allowed.

Table 4.5 SFR List (5)

Address	Register	Symbol	Reset Value
0000E0h			
0000E1h	CAN0 Receive Interrupt Control Register	C0RIC	XXXX X000b
0000E2h			
0000E3h			
0000E4h			
0000E5h			
0000E6h			
0000E7h			
0000E8h			
0000E9h			
0000EAh			
0000EBh			
0000ECh			
0000EDh			
0000EEh			
0000EFh			
0000F0h	CAN0 Receive FIFO Interrupt Control Register	C0FRIC	XXXX X000b
0000F1h			
0000F2h			
0000F3h			
0000F4h			
0000F5h			
0000F6h			
0000F7h			
0000F8h			
0000F9h			
0000FAh			
0000FBh			
0000FCh	INT8 Interrupt Control Register	INT8IC	XX00 X000b
0000FDh	UART7 Receive Interrupt Control Register	S7RIC	XXXX X000b
0000FEh	INT6 Interrupt Control Register	INT6IC	XX00 X000b
0000FFh	UART8 Receive Interrupt Control Register	S8RIC	XXXX X000b
000100h	Group 1 Time Measurement/Waveform Generation Register 0	G1TM0/G1PO0	XXXXh
000101h			
000102h	Group 1 Time Measurement/Waveform Generation Register 1	G1TM1/G1PO1	XXXXh
000103h			
000104h	Group 1 Time Measurement/Waveform Generation Register 2	G1TM2/G1PO2	XXXXh
000105h			
000106h	Group 1 Time Measurement/Waveform Generation Register 3	G1TM3/G1PO3	XXXXh
000107h			

X: Undefined
Blanks are reserved. No access is allowed.

Table 4.6 SFR List (6)

Address	Register	Symbol	Reset Value
000108h	Group 1 Time Measurement/Waveform Generation Register 4	G1TM4/G1PO4	XXXXh
000109h			
00010Ah	Group 1 Time Measurement/Waveform Generation Register 5	G1TM5/G1PO5	XXXXh
00010Bh			
00010Ch	Group 1 Time Measurement/Waveform Generation Register 6	G1TM6/G1PO6	XXXXh
00010Dh			
00010Eh	Group 1 Time Measurement/Waveform Generation Register 7	G1TM7/G1PO7	XXXXh
00010Fh			
000110h	Group 1 Waveform Generation Control Register 0	G1POCR0	0000 X000b
000111h	Group 1 Waveform Generation Control Register 1	G1POCR1	0X00 X000b
000112h	Group 1 Waveform Generation Control Register 2	G1POCR2	0X00 X000b
000113h	Group 1 Waveform Generation Control Register 3	G1POCR3	0X00 X000b
000114h	Group 1 Waveform Generation Control Register 4	G1POCR4	0X00 X000b
000115h	Group 1 Waveform Generation Control Register 5	G1POCR5	0X00 X000b
000116h	Group 1 Waveform Generation Control Register 6	G1POCR6	0X00 X000b
000117h	Group 1 Waveform Generation Control Register 7	G1POCR7	0X00 X000b
000118h	Group 1 Time Measurement Control Register 0	G1TMCR0	00h
000119h	Group 1 Time Measurement Control Register 1	G1TMCR1	00h
00011Ah	Group 1 Time Measurement Control Register 2	G1TMCR2	00h
00011Bh	Group 1 Time Measurement Control Register 3	G1TMCR3	00h
00011Ch	Group 1 Time Measurement Control Register 4	G1TMCR4	00h
00011Dh	Group 1 Time Measurement Control Register 5	G1TMCR5	00h
00011Eh	Group 1 Time Measurement Control Register 6	G1TMCR6	00h
00011Fh	Group 1 Time Measurement Control Register 7	G1TMCR7	00h
000120h	Group 1 Base Timer Register	G1BT	XXXXh
000121h			
000122h	Group 1 Base Timer Control Register 0	G1BCR0	00h
000123h	Group 1 Base Timer Control Register 1	G1BCR1	0000 0000b
000124h	Group 1 Time Measurement Prescaler Register 6	G1TPR6	00h
000125h	Group 1 Time Measurement Prescaler Register 7	G1TPR7	00h
000126h	Group 1 Function Enable Register	G1FE	00h
000127h	Group 1 Function Select Register	G1FS	00h
000128h			
000129h			
00012Ah			
00012Bh			
00012Ch			
00012Dh			
00012Eh			
00012Fh			
000130h to 00013Fh			

X: Undefined

Blanks are reserved. No access is allowed.

Table 4.7 SFR List (7)

Address	Register	Symbol	Reset Value
000140h	Group 2 Waveform Generation Register 0	G2PO0	XXXXh
000141h			
000142h	Group 2 Waveform Generation Register 1	G2PO1	XXXXh
000143h			
000144h	Group 2 Waveform Generation Register 2	G2PO2	XXXXh
000145h			
000146h	Group 2 Waveform Generation Register 3	G2PO3	XXXXh
000147h			
000148h	Group 2 Waveform Generation Register 4	G2PO4	XXXXh
000149h			
00014Ah	Group 2 Waveform Generation Register 5	G2PO5	XXXXh
00014Bh			
00014Ch	Group 2 Waveform Generation Register 6	G2PO6	XXXXh
00014Dh			
00014Eh	Group 2 Waveform Generation Register 7	G2PO7	XXXXh
00014Fh			
000150h	Group 2 Waveform Generation Control Register 0	G2POCR0	0000 0000b
000151h	Group 2 Waveform Generation Control Register 1	G2POCR1	0000 0000b
000152h	Group 2 Waveform Generation Control Register 2	G2POCR2	0000 0000b
000153h	Group 2 Waveform Generation Control Register 3	G2POCR3	0000 0000b
000154h	Group 2 Waveform Generation Control Register 4	G2POCR4	0000 0000b
000155h	Group 2 Waveform Generation Control Register 5	G2POCR5	0000 0000b
000156h	Group 2 Waveform Generation Control Register 6	G2POCR6	0000 0000b
000157h	Group 2 Waveform Generation Control Register 7	G2POCR7	0000 0000b
000158h			
000159h			
00015Ah			
00015Bh			
00015Ch			
00015Dh			
00015Eh			
00015Fh			
000160h	Group 2 Base Timer Register	G2BT	XXXXh
000161h			
000162h	Group 2 Base Timer Control Register 0	G2BCR0	00h
000163h	Group 2 Base Timer Control Register 1	G2BCR1	0000 0000b
000164h	Base Timer Start Register	BTSR	XXXX 0000b
000165h			
000166h	Group 2 Function Enable Register	G2FE	00h
000167h	Group 2 RTP Output Buffer Register	G2RTP	00h
000168h			
000169h			
00016Ah	Group 2 Serial Interface Mode Register	G2MR	00XX X000b
00016Bh	Group 2 Serial Interface Control Register	G2CR	0000 X110b
00016Ch	Group 2 SI/O Transmit Buffer Register	G2TB	XXXXh
00016Dh			
00016Eh	Group 2 SI/O Receive Buffer Register	G2RB	XXXXh
00016Fh			

X: Undefined
Blanks are reserved. No access is allowed.

Table 4.8 SFR List (8)

Address	Register	Symbol	Reset Value
000170h	Group 2 IEBus Address Register	IEAR	XXXXh
000171h			
000172h	Group 2 IEBus Control Register	IECR	00XX X000b
000173h	Group 2 IEBus Transmit Interrupt Source Detect Register	IETIF	XXX0 0000b
000174h	Group 2 IEBus Receive Interrupt Source Detect Register	IERIF	XXX0 0000b
000175h			
000176h			
000177h			
000178h			
000179h			
00017Ah			
00017Bh			
00017Ch			
00017Dh			
00017Eh			
00017Fh			
000180h	Group 0 Time Measurement/Waveform Generation Register 0	G0TM0/G0PO0	XXXXh
000181h			
000182h	Group 0 Time Measurement/Waveform Generation Register 1	G0TM1/G0PO1	XXXXh
000183h			
000184h	Group 0 Time Measurement/Waveform Generation Register 2	G0TM2/G0PO2	XXXXh
000185h			
000186h	Group 0 Time Measurement/Waveform Generation Register 3	G0TM3/G0PO3	XXXXh
000187h			
000188h	Group 0 Time Measurement/Waveform Generation Register 4	G0TM4/G0PO4	XXXXh
000189h			
00018Ah	Group 0 Time Measurement/Waveform Generation Register 5	G0TM5/G0PO5	XXXXh
00018Bh			
00018Ch	Group 0 Time Measurement/Waveform Generation Register 6	G0TM6/G0PO6	XXXXh
00018Dh			
00018Eh	Group 0 Time Measurement/Waveform Generation Register 7	G0TM7/G0PO7	XXXXh
00018Fh			
000190h	Group 0 Waveform Generation Control Register 0	G0POCR0	0000 X000b
000191h	Group 0 Waveform Generation Control Register 1	G0POCR1	0X00 X000b
000192h	Group 0 Waveform Generation Control Register 2	G0POCR2	0X00 X000b
000193h	Group 0 Waveform Generation Control Register 3	G0POCR3	0X00 X000b
000194h	Group 0 Waveform Generation Control Register 4	G0POCR4	0X00 X000b
000195h	Group 0 Waveform Generation Control Register 5	G0POCR5	0X00 X000b
000196h	Group 0 Waveform Generation Control Register 6	G0POCR6	0X00 X000b
000197h	Group 0 Waveform Generation Control Register 7	G0POCR7	0X00 X000b
000198h	Group 0 Time Measurement Control Register 0	G0TMCR0	00h
000199h	Group 0 Time Measurement Control Register 1	G0TMCR1	00h
00019Ah	Group 0 Time Measurement Control Register 2	G0TMCR2	00h
00019Bh	Group 0 Time Measurement Control Register 3	G0TMCR3	00h
00019Ch	Group 0 Time Measurement Control Register 4	G0TMCR4	00h
00019Dh	Group 0 Time Measurement Control Register 5	G0TMCR5	00h
00019Eh	Group 0 Time Measurement Control Register 6	G0TMCR6	00h
00019Fh	Group 0 Time Measurement Control Register 7	G0TMCR7	00h

X: Undefined
Blanks are reserved. No access is allowed.

Table 4.9 SFR List (9)

Address	Register	Symbol	Reset Value
0001A0h	Group 0 Base Timer Register	G0BT	XXXXh
0001A1h			
0001A2h	Group 0 Base Timer Control Register 0	G0BCR0	00h
0001A3h	Group 0 Base Timer Control Register 1	G0BCR1	0000 0000b
0001A4h	Group 0 Time Measurement Prescaler Register 6	G0TPR6	00h
0001A5h	Group 0 Time Measurement Prescaler Register 7	G0TPR7	00h
0001A6h	Group 0 Function Enable Register	G0FE	00h
0001A7h	Group 0 Function Select Register	G0FS	00h
0001A8h			
0001A9h			
0001AAh			
0001ABh			
0001ACh			
0001ADh			
0001AEh			
0001AFh			
0001B0h			
0001B1h			
0001B2h			
0001B3h			
0001B4h			
0001B5h			
0001B6h			
0001B7h			
0001B8h			
0001B9h			
0001BAh			
0001BBh			
0001BCh			
0001BDh			
0001BEh			
0001BFh			
0001C0h			
0001C1h			
0001C2h			
0001C3h			
0001C4h	UART5 Special Mode Register 4	U5SMR4	00h
0001C5h	UART5 Special Mode Register 3	U5SMR3	00h
0001C6h	UART5 Special Mode Register 2	U5SMR2	00h
0001C7h	UART5 Special Mode Register	U5SMR	00h
0001C8h	UART5 Transmit/Receive Mode Register	U5MR	00h
0001C9h	UART5 Bit Rate Register	U5BRG	XXh
0001CAh	UART5 Transmit Buffer Register	U5TB	XXXXh
0001CBh			
0001CCh	UART5 Transmit/Receive Control Register 0	U5C0	0000 1000b
0001CDh	UART5 Transmit/Receive Control Register 1	U5C1	0000 0010b
0001CEh	UART5 Receive Buffer Register	U5RB	XXXXh
0001CFh			

X: Undefined
Blanks are reserved. No access is allowed.

Table 4.10 SFR List (10)

Address	Register	Symbol	Reset Value
0001D0h			
0001D1h			
0001D2h			
0001D3h			
0001D4h	UART6 Special Mode Register 4	U6SMR4	00h
0001D5h	UART6 Special Mode Register 3	U6SMR3	00h
0001D6h	UART6 Special Mode Register 2	U6SMR2	00h
0001D7h	UART6 Special Mode Register	U6SMR	00h
0001D8h	UART6 Transmit/Receive Mode Register	U6MR	00h
0001D9h	UART6 Bit Rate Register	U6BRG	XXh
0001DAh	UART6 Transmit Buffer Register	U6TB	XXXXh
0001DBh			
0001DCh	UART6 Transmit/Receive Control Register 0	U6C0	0000 1000b
0001DDh	UART6 Transmit/Receive Control Register 1	U6C1	0000 0010b
0001DEh	UART6 Receive Buffer Register	U6RB	XXXXh
0001DFh			
0001E0h	UART7 Transmit/Receive Mode Register	U7MR	00h
0001E1h	UART7 Bit Rate Register	U7BRG	XXh
0001E2h	UART7 Transmit Buffer Register	U7TB	XXXXh
0001E3h			
0001E4h	UART7 Transmit/Receive Control Register 0	U7C0	00X0 1000b
0001E5h	UART7 Transmit/Receive Control Register 1	U7C1	XXXX 0010b
0001E6h	UART7 Receive Buffer Register	U7RB	XXXXh
0001E7h			
0001E8h	UART8 Transmit/Receive Mode Register	U8MR	00h
0001E9h	UART8 Bit Rate Register	U8BRG	XXh
0001EAh	UART8 Transmit Buffer Register	U8TB	XXXXh
0001EBh			
0001ECh	UART8 Transmit/Receive Control Register 0	U8C0	00X0 1000b
0001EDh	UART8 Transmit/Receive Control Register 1	U8C1	XXXX 0010b
0001EEh	UART8 Receive Buffer Register	U8RB	XXXXh
0001EFh			
0001F0h	UART7, UART8 Transmit/Receive Control Register 2	U78CON	X000 0000b
0001F1h			
0001F2h			
0001F3h			
0001F4h			
0001F5h			
0001F6h			
0001F7h			
0001F8h			
0001F9h			
0001FAh			
0001FBh			
0001FCh			
0001FDh			
0001FEh			
0001FFh			

X: Undefined
Blanks are reserved. No access is allowed.

Table 4.11 SFR List (11)

Address	Register	Symbol	Reset Value
000200h to 0002BFh			
0002C0h	X0 Register/Y0 Register	X0R/Y0R	XXXXh
0002C1h			
0002C2h	X1 Register/Y1 Register	X1R/Y1R	XXXXh
0002C3h			
0002C4h	X2 Register/Y2 Register	X2R/Y2R	XXXXh
0002C5h			
0002C6h	X3 Register/Y3 Register	X3R/Y3R	XXXXh
0002C7h			
0002C8h	X4 Register/Y4 Register	X4R/Y4R	XXXXh
0002C9h			
0002CAh	X5 Register/Y5 Register	X5R/Y5R	XXXXh
0002CBh			
0002CCh	X6 Register/Y6 Register	X6R/Y6R	XXXXh
0002CDh			
0002CEh	X7 Register/Y7 Register	X7R/Y7R	XXXXh
0002CFh			
0002D0h	X8 Register/Y8 Register	X8R/Y8R	XXXXh
0002D1h			
0002D2h	X9 Register/Y9 Register	X9R/Y9R	XXXXh
0002D3h			
0002D4h	X10 Register/Y10 Register	X10R/Y10R	XXXXh
0002D5h			
0002D6h	X11 Register/Y11 Register	X11R/Y11R	XXXXh
0002D7h			
0002D8h	X12 Register/Y12 Register	X12R/Y12R	XXXXh
0002D9h			
0002DAh	X13 Register/Y13 Register	X13R/Y13R	XXXXh
0002DBh			
0002DCh	X14 Register/Y14 Register	X14R/Y14R	XXXXh
0002DDh			
0002DEh	X15 Register/Y15 Register	X15R/Y15R	XXXXh
0002DFh			
0002E0h	X-Y Control Register	XYC	XXXX XX00b
0002E1h			
0002E2h			
0002E3h			
0002E4h	UART1 Special Mode Register 4	U1SMR4	00h
0002E5h	UART1 Special Mode Register 3	U1SMR3	00h
0002E6h	UART1 Special Mode Register 2	U1SMR2	00h
0002E7h	UART1 Special Mode Register	U1SMR	00h
0002E8h	UART1 Transmit/Receive Mode Register	U1MR	00h
0002E9h	UART1 Bit Rate Register	U1BRG	XXh
0002EAh	UART1 Transmit Buffer Register	U1TB	XXXXh
0002EBh			
0002ECh	UART1 Transmit/Receive Control Register 0	U1C0	0000 1000b
0002EDh	UART1 Transmit/Receive Control Register 1	U1C1	0000 0010b
0002EEh	UART1 Receive Buffer Register	U1RB	XXXXh
0002EFh			

X: Undefined
Blanks are reserved. No access is allowed.

Table 4.12 SFR List (12)

Address	Register	Symbol	Reset Value
0002F0h			
0002F1h			
0002F2h			
0002F3h			
0002F4h	UART4 Special Mode Register 4	U4SMR4	00h
0002F5h	UART4 Special Mode Register 3	U4SMR3	00h
0002F6h	UART4 Special Mode Register 2	U4SMR2	00h
0002F7h	UART4 Special Mode Register	U4SMR	00h
0002F8h	UART4 Transmit/Receive Mode Register	U4MR	00h
0002F9h	UART4 Bit Rate Register	U4BRG	XXh
0002FAh	UART4 Transmit Buffer Register	U4TB	XXXXh
0002FBh			
0002FCh	UART4 Transmit/Receive Control Register 0	U4C0	0000 1000b
0002FDh	UART4 Transmit/Receive Control Register 1	U4C1	0000 0010b
0002FEh	UART4 Receive Buffer Register	U4RB	XXXXh
0002FFh			
000300h	Count Start Register for Timers B3, B4, and B5	TBSR	000X XXXXb
000301h			
000302h	Timer A1-1 Register	TA11	XXXXh
000303h			
000304h	Timer A2-1 Register	TA21	XXXXh
000305h			
000306h	Timer A4-1 Register	TA41	XXXXh
000307h			
000308h	Three-phase PWM Control Register 0	INVC0	00h
000309h	Three-phase PWM Control Register 1	INVC1	00h
00030Ah	Three-phase Output Buffer Register 0	IDB0	XX11 1111b
00030Bh	Three-phase Output Buffer Register 1	IDB1	XX11 1111b
00030Ch	Dead Time Timer	DTT	XXh
00030Dh	Timer B2 Interrupt Generating Frequency Set Counter	ICTB2	XXh
00030Eh			
00030Fh			
000310h	Timer B3 Register	TB3	XXXXh
000311h			
000312h	Timer B4 Register	TB4	XXXXh
000313h			
000314h	Timer B5 Register	TB5	XXXXh
000315h			
000316h			
000317h			
000318h			
000319h			
00031Ah			
00031Bh	Timer B3 Mode Register	TB3MR	00XX 0000b
00031Ch	Timer B4 Mode Register	TB4MR	00XX 0000b
00031Dh	Timer B5 Mode Register	TB5MR	00XX 0000b
00031Eh			
00031Fh			

X: Undefined
Blanks are reserved. No access is allowed.

Table 4.13 SFR List (13)

Address	Register	Symbol	Reset Value
000320h			
000321h			
000322h			
000323h			
000324h	UART3 Special Mode Register 4	U3SMR4	00h
000325h	UART3 Special Mode Register 3	U3SMR3	00h
000326h	UART3 Special Mode Register 2	U3SMR2	00h
000327h	UART3 Special Mode Register	U3SMR	00h
000328h	UART3 Transmit/Receive Mode Register	U3MR	00h
000329h	UART3 Bit Rate Register	U3BRG	XXh
00032Ah	UART3 Transmit Buffer Register	U3TB	XXXXh
00032Bh			
00032Ch	UART3 Transmit/Receive Control Register 0	U3C0	0000 1000b
00032Dh	UART3 Transmit/Receive Control Register 1	U3C1	0000 0010b
00032Eh	UART3 Receive Buffer Register	U3RB	XXXXh
00032Fh			
000330h			
000331h			
000332h			
000333h			
000334h	UART2 Special Mode Register 4	U2SMR4	00h
000335h	UART2 Special Mode Register 3	U2SMR3	00h
000336h	UART2 Special Mode Register 2	U2SMR2	00h
000337h	UART2 Special Mode Register	U2SMR	00h
000338h	UART2 Transmit/Receive Mode Register	U2MR	00h
000339h	UART2 Bit Rate Register	U2BRG	XXh
00033Ah	UART2 Transmit Buffer Register	U2TB	XXXXh
00033Bh			
00033Ch	UART2 Transmit/Receive Control Register 0	U2C0	0000 1000b
00033Dh	UART2 Transmit/Receive Control Register 1	U2C1	0000 0010b
00033Eh	UART2 Receive Buffer Register	U2RB	XXXXh
00033Fh			
000340h	Count Start Register	TABSR	0000 0000b
000341h	Clock Prescaler Reset Register	CPSRF	0XXX XXXXb
000342h	One-shot Start Register	ONSF	0000 0000b
000343h	Trigger Select Register	TRGSR	0000 0000b
000344h	Increment/Decrement Counting Select Register	UDF	0000 0000b
000345h			
000346h	Timer A0 Register	TA0	XXXXh
000347h			
000348h	Timer A1 Register	TA1	XXXXh
000349h			
00034Ah	Timer A2 Register	TA2	XXXXh
00034Bh			
00034Ch	Timer A3 Register	TA3	XXXXh
00034Dh			
00034Eh	Timer A4 Register	TA4	XXXXh
00034Fh			

X: Undefined
Blanks are reserved. No access is allowed.

Table 4.14 SFR List (14)

Address	Register	Symbol	Reset Value
000350h	Timer B0 Register	TB0	XXXXh
000351h			
000352h	Timer B1 Register	TB1	XXXXh
000353h			
000354h	Timer B2 Register	TB2	XXXXh
000355h			
000356h	Timer A0 Mode Register	TA0MR	0000 0000b
000357h	Timer A1 Mode Register	TA1MR	0000 0000b
000358h	Timer A2 Mode Register	TA2MR	0000 0000b
000359h	Timer A3 Mode Register	TA3MR	0000 0000b
00035Ah	Timer A4 Mode Register	TA4MR	0000 0000b
00035Bh	Timer B0 Mode Register	TB0MR	00XX 0000b
00035Ch	Timer B1 Mode Register	TB1MR	00XX 0000b
00035Dh	Timer B2 Mode Register	TB2MR	00XX 0000b
00035Eh	Timer B2 Special Mode Register	TB2SC	XXXX XXX0b
00035Fh	Count Source Prescaler Register	TCSPR	0000 0000b
000360h			
000361h			
000362h			
000363h			
000364h	UART0 Special Mode Register 4	U0SMR4	00h
000365h	UART0 Special Mode Register 3	U0SMR3	00h
000366h	UART0 Special Mode Register 2	U0SMR2	00h
000367h	UART0 Special Mode Register	U0SMR	00h
000368h	UART0 Transmit/Receive Mode Register	U0MR	00h
000369h	UART0 Bit Rate Register	U0BRG	XXh
00036Ah	UART0 Transmit Buffer Register	U0TB	XXXXh
00036Bh			
00036Ch	UART0 Transmit/Receive Control Register 0	U0C0	0000 1000b
00036Dh	UART0 Transmit/Receive Control Register 1	U0C1	0000 0010b
00036Eh	UART0 Receive Buffer Register	U0RB	XXXXh
00036Fh			
000370h			
000371h			
000372h			
000373h			
000374h			
000375h			
000376h			
000377h			
000378h			
000379h			
00037Ah			
00037Bh			
00037Ch	CRC Data Register	CRCD	XXXXh
00037Dh			
00037Eh	CRC Input Register	CRCIN	XXh
00037Fh			

X: Undefined
Blanks are reserved. No access is allowed.

Table 4.15 SFR List (15)

Address	Register	Symbol	Reset Value
000380h	A/D0 Register 0	AD00	00XXh
000381h			
000382h	A/D0 Register 1	AD01	00XXh
000383h			
000384h	A/D0 Register 2	AD02	00XXh
000385h			
000386h	A/D0 Register 3	AD03	00XXh
000387h			
000388h	A/D0 Register 4	AD04	00XXh
000389h			
00038Ah	A/D0 Register 5	AD05	00XXh
00038Bh			
00038Ch	A/D0 Register 6	AD06	00XXh
00038Dh			
00038Eh	A/D0 Register 7	AD07	00XXh
00038Fh			
000390h			
000391h			
000392h	A/D0 Control Register 4	AD0CON4	XXXX 00XXb
000393h			
000394h	A/D0 Control Register 2	AD0CON2	XX0X X000b
000395h	A/D0 Control Register 3	AD0CON3	XXXX X000b
000396h	A/D0 Control Register 0	AD0CON0	00h
000397h	A/D0 Control Register 1	AD0CON1	00h
000398h	D/A Register 0	DA0	XXh
000399h			
00039Ah	D/A Register 1	DA1	XXh
00039Bh			
00039Ch	D/A Control Register	DACON	XXXX XX00b
00039Dh			
00039Eh			
00039Fh			
0003A0h			
0003A1h			
0003A2h			
0003A3h			
0003A4h			
0003A5h			
0003A6h			
0003A7h			
0003A8h			
0003A9h			
0003AAh			
0003ABh			
0003ACh			
0003ADh			
0003AEh			
0003AFh			

X: Undefined
Blanks are reserved. No access is allowed.

Table 4.16 SFR List (16)

Address	Register	Symbol	Reset Value
0003B0h			
0003B1h			
0003B2h			
0003B3h			
0003B4h			
0003B5h			
0003B6h			
0003B7h			
0003B8h			
0003B9h			
0003BAh			
0003BBh			
0003BCh			
0003BDh			
0003BEh			
0003BFh			
0003C0h	Port P0 Register	P0	XXh
0003C1h	Port P1 Register	P1	XXh
0003C2h	Port P0 Direction Register	PD0	0000 0000b
0003C3h	Port P1 Direction Register	PD1	0000 0000b
0003C4h	Port P2 Register	P2	XXh
0003C5h	Port P3 Register	P3	XXh
0003C6h	Port P2 Direction Register	PD2	0000 0000b
0003C7h	Port P3 Direction Register	PD3	0000 0000b
0003C8h	Port P4 Register	P4	XXh
0003C9h	Port P5 Register	P5	XXh
0003CAh	Port P4 Direction Register	PD4	0000 0000b
0003CBh	Port P5 Direction Register	PD5	0000 0000b
0003CCh	Port P6 Register	P6	XXh
0003CDh	Port P7 Register	P7	XXh
0003CEh	Port P6 Direction Register	PD6	0000 0000b
0003CFh	Port P7 Direction Register	PD7	0000 0000b
0003D0h	Port P8 Register	P8	XXh
0003D1h	Port P9 Register	P9	XXh
0003D2h	Port P8 Direction Register	PD8	00X0 0000b
0003D3h	Port P9 Direction Register	PD9	0000 0000b
0003D4h	Port P10 Register	P10	XXh
0003D5h	Port P11 Register	P11	XXh
0003D6h	Port P10 Direction Register	PD10	0000 0000b
0003D7h	Port P11 Direction Register	PD11	XXX0 0000b
0003D8h	Port P12 Register	P12	XXh
0003D9h	Port P13 Register	P13	XXh
0003DAh	Port P12 Direction Register	PD12	0000 0000b
0003DBh	Port P13 Direction Register	PD13	0000 0000b
0003DCh	Port P14 Register	P14	XXh
0003DDh	Port P15 Register	P15	XXh
0003DEh	Port P14 Direction Register	PD14	X000 0000b
0003DFh	Port P15 Direction Register	PD15	0000 0000b

X: Undefined
Blanks are reserved. No access is allowed.

Table 4.17 SFR List (17)

Address	Register	Symbol	Reset Value
0003E0h			
0003E1h			
0003E2h			
0003E3h			
0003E4h			
0003E5h			
0003E6h			
0003E7h			
0003E8h			
0003E9h			
0003EAh			
0003EBh			
0003ECh			
0003EDh			
0003EEh			
0003EFh			
0003F0h	Pull-up Control Register 0	PUR0	0000 0000b
0003F1h	Pull-up Control Register 1	PUR1	XXXX X0XXb
0003F2h	Pull-up Control Register 2	PUR2	000X XXXXb
0003F3h	Pull-up Control Register 3	PUR3	0000 0000b
0003F4h	Pull-up Control Register 4	PUR4	XXXX 0000b
0003F5h			
0003F6h			
0003F7h			
0003F8h			
0003F9h			
0003FAh			
0003FBh			
0003FCh			
0003FDh			
0003FEh			
0003FFh	Port Control Register	PCR	1XXX XXX0b

X: Undefined
Blanks are reserved. No access is allowed.

Table 4.18 SFR List (18)

Address	Register	Symbol	Reset Value
040000h	Flash Memory Control Register 0	FMR0	0X01 XX00b
040001h	Flash Memory Status Register 0	FMSR0	1000 0000b
040002h			
040003h			
040004h			
040005h			
040006h			
040007h			
040008h	Flash Register Protection Unlock Register 0	FPR0	00h
040009h	Flash Memory Control Register 1	FMR1	0000 0010b
04000Ah	Block Protect Bit Monitor Register 0	FBPM0	??X? ????b (1)
04000Bh	Block Protect Bit Monitor Register 1	FBPM1	XXX? ????b (1)
04000Ch			
04000Dh			
04000Eh			
04000Fh			
040010h			
040011h	Block Protect Bit Monitor Register 2	FBPM2	???? ????b (1)
040012h			
040013h			
040014h			
040015h			
040016h			
040017h			
040018h			
040019h			
04001Ah			
04001Bh			
04001Ch			
04001Dh			
04001Eh			
04001Fh			
040020h	PLL Control Register 0	PLC0	0000 0001b
040021h	PLL Control Register 1	PLC1	0001 1111b
040022h			
040023h			
040024h			
040025h			
040026h			
040027h			
040028h			
040029h			
04002Ah			
04002Bh			
04002Ch			
04002Dh			
04002Eh			
04002Fh			

X: Undefined
Blanks are reserved. No access is allowed.

Note:
1. The reset value reflects the value of the protect bit for each block in the flash memory.

Table 4.19 SFR List (19)

Address	Register	Symbol	Reset Value
040030h to 04003Fh			
040040h			
040041h			
040042h			
040043h			
040044h	Processor Mode Register 0	PM0	1000 0000b
040045h			
040046h	System Clock Control Register 0	CM0	0000 1000b
040047h	System Clock Control Register 1	CM1	0010 0000b
040048h	Processor Mode Register 3	PM3	00h
040049h			
04004Ah	Protect Register	PRCR	XXXX X000b
04004Bh			
04004Ch	Protect Register 3	PRCR3	0000 0000b
04004Dh	Oscillator Stop Detection Register	CM2	00h
04004Eh			
04004Fh			
040050h			
040051h			
040052h			
040053h	Processor Mode Register 2	PM2	00h
040054h			
040055h			
040056h			
040057h			
040058h			
040059h			
04005Ah	Low Speed Mode Clock Control Register	CM3	XXXX XX00b
04005Bh			
04005Ch			
04005Dh			
04005Eh			
04005Fh			
040060h	Voltage Regulator Control Register	VRCR	0000 0000b
040061h			
040062h			
040063h			
040064h			
040065h			
040066h			
040067h			
040068h to 040093h			

X: Undefined
Blanks are reserved. No access is allowed.

Table 4.20 SFR List (20)

Address	Register	Symbol	Reset Value
040094h			
040095h			
040096h			
040097h	Three-phase Output Buffer Control Register	IOBC	0XXX XXXXb
040098h	Input Function Select Register 0	IFS0	X000 0000b
040099h	Input Function Select Register 1	IFS1	XXXX X0X0b
04009Ah	Input Function Select Register 2	IFS2	0000 00X0b
04009Bh	Input Function Select Register 3	IFS3	XXXX XX00b
04009Ch			
04009Dh			
04009Eh			
04009Fh			
0400A0h	Port P0_0 Function Select Register	P0_0S	0XXX X000b
0400A1h	Port P1_0 Function Select Register	P1_0S	XXXX X000b
0400A2h	Port P0_1 Function Select Register	P0_1S	0XXX X000b
0400A3h	Port P1_1 Function Select Register	P1_1S	XXXX X000b
0400A4h	Port P0_2 Function Select Register	P0_2S	0XXX X000b
0400A5h	Port P1_2 Function Select Register	P1_2S	XXXX X000b
0400A6h	Port P0_3 Function Select Register	P0_3S	0XXX X000b
0400A7h	Port P1_3 Function Select Register	P1_3S	XXXX X000b
0400A8h	Port P0_4 Function Select Register	P0_4S	0XXX X000b
0400A9h	Port P1_4 Function Select Register	P1_4S	XXXX X000b
0400AAh	Port P0_5 Function Select Register	P0_5S	0XXX X000b
0400ABh	Port P1_5 Function Select Register	P1_5S	XXXX X000b
0400ACh	Port P0_6 Function Select Register	P0_6S	0XXX X000b
0400ADh	Port P1_6 Function Select Register	P1_6S	XXXX X000b
0400AEh	Port P0_7 Function Select Register	P0_7S	0XXX X000b
0400AFh	Port P1_7 Function Select Register	P1_7S	XXXX X000b
0400B0h	Port P2_0 Function Select Register	P2_0S	0XXX X000b
0400B1h	Port P3_0 Function Select Register	P3_0S	XXXX X000b
0400B2h	Port P2_1 Function Select Register	P2_1S	0XXX X000b
0400B3h	Port P3_1 Function Select Register	P3_1S	XXXX X000b
0400B4h	Port P2_2 Function Select Register	P2_2S	0XXX X000b
0400B5h	Port P3_2 Function Select Register	P3_2S	XXXX X000b
0400B6h	Port P2_3 Function Select Register	P2_3S	0XXX X000b
0400B7h	Port P3_3 Function Select Register	P3_3S	XXXX X000b
0400B8h	Port P2_4 Function Select Register	P2_4S	0XXX X000b
0400B9h	Port P3_4 Function Select Register	P3_4S	XXXX X000b
0400BAh	Port P2_5 Function Select Register	P2_5S	0XXX X000b
0400BBh	Port P3_5 Function Select Register	P3_5S	XXXX X000b
0400BCh	Port P2_6 Function Select Register	P2_6S	0XXX X000b
0400BDh	Port P3_6 Function Select Register	P3_6S	XXXX X000b
0400BEh	Port P2_7 Function Select Register	P2_7S	0XXX X000b
0400BFh	Port P3_7 Function Select Register	P3_7S	XXXX X000b

X: Undefined
Blanks are reserved. No access is allowed.

Table 4.21 SFR List (21)

Address	Register	Symbol	Reset Value
0400C0h	Port P4_0 Function Select Register	P4_0S	X0XX X000b
0400C1h	Port P5_0 Function Select Register	P5_0S	XXXX X000b
0400C2h	Port P4_1 Function Select Register	P4_1S	X0XX X000b
0400C3h	Port P5_1 Function Select Register	P5_1S	XXXX X000b
0400C4h	Port P4_2 Function Select Register	P4_2S	X0XX X000b
0400C5h	Port P5_2 Function Select Register	P5_2S	XXXX X000b
0400C6h	Port P4_3 Function Select Register	P4_3S	X0XX X000b
0400C7h	Port P5_3 Function Select Register	P5_3S	XXXX X000b
0400C8h	Port P4_4 Function Select Register	P4_4S	X0XX X000b
0400C9h	Port P5_4 Function Select Register	P5_4S	X0XX X000b
0400CAh	Port P4_5 Function Select Register	P4_5S	X0XX X000b
0400CBh	Port P5_5 Function Select Register	P5_5S	X0XX X000b
0400CCh	Port P4_6 Function Select Register	P4_6S	X0XX X000b
0400CDh	Port P5_6 Function Select Register	P5_6S	X0XX X000b
0400CEh	Port P4_7 Function Select Register	P4_7S	X0XX X000b
0400CFh	Port P5_7 Function Select Register	P5_7S	X0XX X000b
0400D0h	Port P6_0 Function Select Register	P6_0S	X0XX X000b
0400D1h	Port P7_0 Function Select Register	P7_0S	X0XX X000b
0400D2h	Port P6_1 Function Select Register	P6_1S	X0XX X000b
0400D3h	Port P7_1 Function Select Register	P7_1S	X0XX X000b
0400D4h	Port P6_2 Function Select Register	P6_2S	X0XX X000b
0400D5h	Port P7_2 Function Select Register	P7_2S	X0XX X000b
0400D6h	Port P6_3 Function Select Register	P6_3S	X0XX X000b
0400D7h	Port P7_3 Function Select Register	P7_3S	X0XX X000b
0400D8h	Port P6_4 Function Select Register	P6_4S	X0XX X000b
0400D9h	Port P7_4 Function Select Register	P7_4S	X0XX X000b
0400DAh	Port P6_5 Function Select Register	P6_5S	X0XX X000b
0400DBh	Port P7_5 Function Select Register	P7_5S	X0XX X000b
0400DCh	Port P6_6 Function Select Register	P6_6S	X0XX X000b
0400DDh	Port P7_6 Function Select Register	P7_6S	X0XX X000b
0400DEh	Port P6_7 Function Select Register	P6_7S	X0XX X000b
0400DFh	Port P7_7 Function Select Register	P7_7S	X0XX X000b
0400E0h	Port P8_0 Function Select Register	P8_0S	X0XX X000b
0400E1h	Port P9_0 Function Select Register	P9_0S	X0XX X000b
0400E2h	Port P8_1 Function Select Register	P8_1S	X0XX X000b
0400E3h	Port P9_1 Function Select Register	P9_1S	X0XX X000b
0400E4h	Port P8_2 Function Select Register	P8_2S	X0XX X000b
0400E5h	Port P9_2 Function Select Register	P9_2S	X0XX X000b
0400E6h	Port P8_3 Function Select Register	P8_3S	X0XX X000b
0400E7h	Port P9_3 Function Select Register	P9_3S	00XX X000b
0400E8h	Port P8_4 Function Select Register	P8_4S	XXXX X000b
0400E9h	Port P9_4 Function Select Register	P9_4S	00XX X000b
0400EAh			
0400EBh	Port P9_5 Function Select Register	P9_5S	00XX X000b
0400ECh	Port P8_6 Function Select Register	P8_6S	XXXX X000b
0400EDh	Port P9_6 Function Select Register	P9_6S	00XX X000b
0400EEh	Port P8_7 Function Select Register	P8_7S	XXXX X000b
0400EFh	Port P9_7 Function Select Register	P9_7S	X0XX X000b

X: Undefined
Blanks are reserved. No access is allowed.

Table 4.22 SFR List (22)

Address	Register	Symbol	Reset Value
0400F0h	Port P10_0 Function Select Register	P10_0S	0XXX X000b
0400F1h	Port P11_0 Function Select Register	P11_0S	X0XX X000b
0400F2h	Port P10_1 Function Select Register	P10_1S	0XXX X000b
0400F3h	Port P11_1 Function Select Register	P11_1S	X0XX X000b
0400F4h	Port P10_2 Function Select Register	P10_2S	0XXX X000b
0400F5h	Port P11_2 Function Select Register	P11_2S	X0XX X000b
0400F6h	Port P10_3 Function Select Register	P10_3S	0XXX X000b
0400F7h	Port P11_3 Function Select Register	P11_3S	X0XX X000b
0400F8h	Port P10_4 Function Select Register	P10_4S	0XXX X000b
0400F9h	Port P11_4 Function Select Register	P11_4S	XXXX X000b
0400FAh	Port P10_5 Function Select Register	P10_5S	0XXX X000b
0400FBh			
0400FCh	Port P10_6 Function Select Register	P10_6S	0XXX X000b
0400FDh			
0400FEh	Port P10_7 Function Select Register	P10_7S	0XXX X000b
0400FFh			
040100h	Port P12_0 Function Select Register	P12_0S	X0XX X000b
040101h	Port P13_0 Function Select Register	P13_0S	XXXX X000b
040102h	Port P12_1 Function Select Register	P12_1S	X0XX X000b
040103h	Port P13_1 Function Select Register	P13_1S	XXXX X000b
040104h	Port P12_2 Function Select Register	P12_2S	X0XX X000b
040105h	Port P13_2 Function Select Register	P13_2S	XXXX X000b
040106h	Port P12_3 Function Select Register	P12_3S	X0XX X000b
040107h	Port P13_3 Function Select Register	P13_3S	XXXX X000b
040108h	Port P12_4 Function Select Register	P12_4S	XXXX X000b
040109h	Port P13_4 Function Select Register	P13_4S	XXXX X000b
04010Ah	Port P12_5 Function Select Register	P12_5S	XXXX X000b
04010Bh	Port P13_5 Function Select Register	P13_5S	XXXX X000b
04010Ch	Port P12_6 Function Select Register	P12_6S	XXXX X000b
04010Dh	Port P13_6 Function Select Register	P13_6S	XXXX X000b
04010Eh	Port P12_7 Function Select Register	P12_7S	XXXX X000b
04010Fh	Port P13_7 Function Select Register	P13_7S	XXXX X000b
040110h			
040111h	Port P15_0 Function Select Register	P15_0S	00XX X000b
040112h			
040113h	Port P15_1 Function Select Register	P15_1S	00XX X000b
040114h			
040115h	Port P15_2 Function Select Register	P15_2S	00XX X000b
040116h	Port P14_3 Function Select Register	P14_3S	XXXX X000b
040117h	Port P15_3 Function Select Register	P15_3S	00XX X000b
040118h	Port P14_4 Function Select Register	P14_4S	XXXX X000b
040119h	Port P15_4 Function Select Register	P15_4S	00XX X000b
04011Ah	Port P14_5 Function Select Register	P14_5S	XXXX X000b
04011Bh	Port P15_5 Function Select Register	P15_5S	00XX X000b
04011Ch	Port P14_6 Function Select Register	P14_6S	XXXX X000b
04011Dh	Port P15_6 Function Select Register	P15_6S	00XX X000b
04011Eh			
04011Fh	Port P15_7 Function Select Register	P15_7S	00XX X000b

X: Undefined
Blanks are reserved. No access is allowed.

Table 4.23 SFR List (23)

Address	Register	Symbol	Reset Value
040120h to 04403Fh			
044040h			
044041h			
044042h			
044043h			
044044h			
044045h			
044046h			
044047h			
044048h			
044049h			
04404Ah			
04404Bh			
04404Ch			
04404Dh			
04404Eh	Watchdog Timer Start Register	WDTS	XXXX XXXXb
04404Fh	Watchdog Timer Control Register	WDC	000X XXXXb
044050h			
044051h			
044052h			
044053h			
044054h			
044055h			
044056h			
044057h			
044058h			
044059h			
04405Ah			
04405Bh			
04405Ch			
04405Dh			
04405Eh			
04405Fh	Protect Register 2	PRCR2	0XXX XXXXb

X: Undefined
Blanks are reserved. No access is allowed.

Table 4.24 SFR List (24)

Address	Register	Symbol	Reset Value
044060h			
044061h			
044062h			
044063h			
044064h			
044065h			
044066h			
044067h			
044068h			
044069h			
04406Ah			
04406Bh			
04406Ch			
04406Dh	External Interrupt Request Source Select Register 1	IFSR1	X0XX X000b
04406Eh			
04406Fh	External Interrupt Request Source Select Register 0	IFSR0	0000 0000b
044070h	DMA0 Request Source Select Register 2	DM0SL2	XX00 0000b
044071h	DMA1 Request Source Select Register 2	DM1SL2	XX00 0000b
044072h	DMA2 Request Source Select Register 2	DM2SL2	XX00 0000b
044073h	DMA3 Request Source Select Register 2	DM3SL2	XX00 0000b
044074h			
044075h			
044076h			
044077h			
044078h	DMA0 Request Source Select Register	DM0SL	XXX0 0000b
044079h	DMA1 Request Source Select Register	DM1SL	XXX0 0000b
04407Ah	DMA2 Request Source Select Register	DM2SL	XXX0 0000b
04407Bh	DMA3 Request Source Select Register	DM3SL	XXX0 0000b
04407Ch			
04407Dh	Wake-up IPL Setting Register 2	RIPL2	XX0X 0000b
04407Eh			
04407Fh	Wake-up IPL Setting Register 1	RIPL1	XX0X 0000b
044080h			
044081h			
044082h			
044083h			
044084h			
044085h			
044086h			
044087h			
044088h			
044089h			
04408Ah			
04408Bh			
04408Ch			
04408Dh			
04408Eh			
04408Fh			

X: Undefined
Blanks are reserved. No access is allowed.

Table 4.25 SFR List (25)

Address	Register	Symbol	Reset Value
044090h to 0443FFh			
044400h	I²C-bus Transmit/Receive Shift Register	I2CTRSR	XXh
044401h			
044402h	I²C-bus Slave Address Register	I2CSAR	00h
044403h	I²C-bus Control Register 0	I2CCR0	0000 0000b
044404h	I²C-bus Clock Control Register	I2CCCR	0000 0000b
044405h	I²C-bus START and STOP Conditions Control Register	I2CSSCR	0001 1010b
044406h	I²C-bus Control Register 1	I2CCR1	0011 0000b
044407h	I²C-bus Control Register 2	I2CCR2	0X00 0000b
044408h	I²C-bus Status Register	I2CSR	0001 000Xb
044409h			
04440Ah			
04440Bh			
04440Ch			
04440Dh			
04440Eh			
04440Fh			
044410h	I²C-bus Mode Register	I2CMR	XXXX 0000b
044411h			
044412h			
044413h			
044414h			
044415h			
044416h			
044417h			
044418h			
044419h			
04441Ah			
04441Bh			
04441Ch			
04441Dh			
04441Eh			
04441Fh			
044420h to 0467FFh			

X: Undefined
Blanks are reserved. No access is allowed.

Table 4.26 SFR List (26)

Address	Register	Symbol	Reset Value
046800h to 047BFFh			
047C00h	CAN0 Mailbox 0: Message Identifier	C0MB0	XXXX XXXXh
047C01h			
047C02h			
047C03h			
047C04h			
047C05h	CAN0 Mailbox 0: Data Length		XXh
047C06h	CAN0 Mailbox 0: Data Field		XXXX XXXX
047C07h			XXXX XXXXh
047C08h			
047C09h			
047C0Ah			
047C0Bh			
047C0Ch			
047C0Dh			
047C0Eh	CAN0 Mailbox 0: Time Stamp		XXXXh
047C0Fh			
047C10h	CAN0 Mailbox 1: Message Identifier	C0MB1	XXXX XXXXh
047C11h			
047C12h			
047C13h			
047C14h			
047C15h	CAN0 Mailbox 1: Data Length		XXh
047C16h	CAN0 Mailbox 1: Data Field		XXXX XXXX
047C17h			XXXX XXXXh
047C18h			
047C19h			
047C1Ah			
047C1Bh			
047C1Ch			
047C1Dh			
047C1Eh	CAN0 Mailbox 1: Time Stamp		XXXXh
047C1Fh			
047C20h	CAN0 Mailbox 2: Message Identifier	C0MB2	XXXX XXXXh
047C21h			
047C22h			
047C23h			
047C24h			
047C25h	CAN0 Mailbox 2: Data Length		XXh
047C26h	CAN0 Mailbox 2: Data Field		XXXX XXXX
047C27h			XXXX XXXXh
047C28h			
047C29h			
047C2Ah			
047C2Bh			
047C2Ch			
047C2Dh			
047C2Eh	CAN0 Mailbox 2: Time Stamp		XXXXh
047C2Fh			

X: Undefined
Blanks are reserved. No access is allowed.

Table 4.27 SFR List (27)

Address	Register	Symbol	Reset Value
047C30h	CAN0 Mailbox 3: Message Identifier	C0MB3	XXXX XXXXh
047C31h			
047C32h			
047C33h			
047C34h			
047C35h	CAN0 Mailbox 3: Data Length		XXh
047C36h	CAN0 Mailbox 3: Data Field		XXXX XXXX
047C37h			XXXX XXXXh
047C38h			
047C39h			
047C3Ah			
047C3Bh			
047C3Ch			
047C3Dh			
047C3Eh	CAN0 Mailbox 3: Time Stamp		XXXXh
047C3Fh			
047C40h	CAN0 Mailbox 4: Message Identifier	C0MB4	XXXX XXXXh
047C41h			
047C42h			
047C43h			
047C44h			
047C45h	CAN0 Mailbox 4: Data Length		XXh
047C46h	CAN0 Mailbox 4: Data Field		XXXX XXXX
047C47h			XXXX XXXXh
047C48h			
047C49h			
047C4Ah			
047C4Bh			
047C4Ch			
047C4Dh			
047C4Eh	CAN0 Mailbox 4: Time Stamp		XXXXh
047C4Fh			
047C50h	CAN0 Mailbox 5: Message Identifier	C0MB5	XXXX XXXXh
047C51h			
047C52h			
047C53h			
047C54h			
047C55h	CAN0 Mailbox 5: Data Length		XXh
047C56h	CAN0 Mailbox 5: Data Field		XXXX XXXX
047C57h			XXXX XXXXh
047C58h			
047C59h			
047C5Ah			
047C5Bh			
047C5Ch			
047C5Dh			
047C5Eh	CAN0 Mailbox 5: Time Stamp		XXXXh
047C5Fh			

X: Undefined
Blanks are reserved. No access is allowed.

Table 4.28 SFR List (28)

Address	Register	Symbol	Reset Value
047C60h	CAN0 Mailbox 6: Message Identifier	C0MB6	XXXX XXXXh
047C61h			
047C62h			
047C63h			
047C64h			
047C65h	CAN0 Mailbox 6: Data Length		XXh
047C66h	CAN0 Mailbox 6: Data Field		XXXX XXXX
047C67h			XXXX XXXXh
047C68h			
047C69h			
047C6Ah			
047C6Bh			
047C6Ch			
047C6Dh			
047C6Eh	CAN0 Mailbox 6: Time Stamp		XXXXh
047C6Fh			
047C70h	CAN0 Mailbox 7: Message Identifier	C0MB7	XXXX XXXXh
047C71h			
047C72h			
047C73h			
047C74h			
047C75h	CAN0 Mailbox 7: Data Length		XXh
047C76h	CAN0 Mailbox 7: Data Field		XXXX XXXX
047C77h			XXXX XXXXh
047C78h			
047C79h			
047C7Ah			
047C7Bh			
047C7Ch			
047C7Dh			
047C7Eh	CAN0 Mailbox 7: Time Stamp		XXXXh
047C7Fh			
047C80h	CAN0 Mailbox 8: Message Identifier	C0MB8	XXXX XXXXh
047C81h			
047C82h			
047C83h			
047C84h			
047C85h	CAN0 Mailbox 8: Data Length		XXh
047C86h	CAN0 Mailbox 8: Data Field		XXXX XXXX
047C87h			XXXX XXXXh
047C88h			
047C89h			
047C8Ah			
047C8Bh			
047C8Ch			
047C8Dh			
047C8Eh	CAN0 Mailbox 8: Time Stamp		XXXXh
047C8Fh			

X: Undefined
Blanks are reserved. No access is allowed.

Table 4.29 SFR List (29)

Address	Register	Symbol	Reset Value
047C90h	CAN0 Mailbox 9: Message Identifier	C0MB9	XXXX XXXXh
047C91h			
047C92h			
047C93h			
047C94h			
047C95h	CAN0 Mailbox 9: Data Length		XXh
047C96h	CAN0 Mailbox 9: Data Field		XXXX XXXX
047C97h			XXXX XXXXh
047C98h			
047C99h			
047C9Ah			
047C9Bh			
047C9Ch			
047C9Dh			
047C9Eh	CAN0 Mailbox 9: Time Stamp		XXXXh
047C9Fh			
047CA0h	CAN0 Mailbox 10: Message Identifier	C0MB10	XXXX XXXXh
047CA1h			
047CA2h			
047CA3h			
047CA4h			
047CA5h	CAN0 Mailbox 10: Data Length		XXh
047CA6h	CAN0 Mailbox 10: Data Field		XXXX XXXX
047CA7h			XXXX XXXXh
047CA8h			
047CA9h			
047CAAh			
047CABh			
047CACh			
047CADh			
047CAEh	CAN0 Mailbox 10: Time Stamp		XXXXh
047CAFh			
047CB0h	CAN0 Mailbox 11: Message Identifier	C0MB11	XXXX XXXXh
047CB1h			
047CB2h			
047CB3h			
047CB4h			
047CB5h	CAN0 Mailbox 11: Data Length		XXh
047CB6h	CAN0 Mailbox 11: Data Field		XXXX XXXX
047CB7h			XXXX XXXXh
047CB8h			
047CB9h			
047CBAh			
047CBBh			
047CBCh			
047CBDh			
047CBEh	CAN0 Mailbox 11: Time Stamp		XXXXh
047CBFh			

X: Undefined
Blanks are reserved. No access is allowed.

Table 4.30 SFR List (30)

Address	Register	Symbol	Reset Value
047CC0h	CAN0 Mailbox 12: Message Identifier	C0MB12	XXXX XXXXh
047CC1h			
047CC2h			
047CC3h			
047CC4h			
047CC5h	CAN0 Mailbox 12: Data Length		XXh
047CC6h	CAN0 Mailbox 12: Data Field		XXXX XXXX
047CC7h			XXXX XXXXh
047CC8h			
047CC9h			
047CCAh			
047CCBh			
047CCCh			
047CCDh			
047CCEh	CAN0 Mailbox 12: Time Stamp		XXXXh
047CCFh			
047CD0h	CAN0 Mailbox 13: Message Identifier	C0MB13	XXXX XXXXh
047CD1h			
047CD2h			
047CD3h			
047CD4h			
047CD5h	CAN0 Mailbox 13: Data Length		XXh
047CD6h	CAN0 Mailbox 13: Data Field		XXXX XXXX
047CD7h			XXXX XXXXh
047CD8h			
047CD9h			
047CDAh			
047CDBh			
047CDCh			
047CDDh			
047CDEh	CAN0 Mailbox 13: Time Stamp		XXXXh
047CDFh			
047CE0h	CAN0 Mailbox 14: Message Identifier	C0MB14	XXXX XXXXh
047CE1h			
047CE2h			
047CE3h			
047CE4h			
047CE5h	CAN0 Mailbox 14: Data Length		XXh
047CE6h	CAN0 Mailbox 14: Data Field		XXXX XXXX
047CE7h			XXXX XXXXh
047CE8h			
047CE9h			
047CEAh			
047CEBh			
047CECh			
047CEDh			
047CEEh	CAN0 Mailbox 14: Time Stamp		XXXXh
047CEFh			

X: Undefined
Blanks are reserved. No access is allowed.

Table 4.31 SFR List (31)

Address	Register	Symbol	Reset Value
047CF0h	CAN0 Mailbox 15: Message Identifier	C0MB15	XXXX XXXXh
047CF1h			
047CF2h			
047CF3h			
047CF4h			
047CF5h	CAN0 Mailbox 15: Data Length		XXh
047CF6h	CAN0 Mailbox 15: Data Field		XXXX XXXX
047CF7h			XXXX XXXXh
047CF8h			
047CF9h			
047CFAh			
047CFBh			
047CFCh			
047CFDh			
047CFEh	CAN0 Mailbox 15: Time Stamp		XXXXh
047CFFh			
047D00h	CAN0 Mailbox 16: Message Identifier	C0MB16	XXXX XXXXh
047D01h			
047D02h			
047D03h			
047D04h			
047D05h	CAN0 Mailbox 16: Data Length		XXh
047D06h	CAN0 Mailbox 16: Data Field		XXXX XXXX
047D07h			XXXX XXXXh
047D08h			
047D09h			
047D0Ah			
047D0Bh			
047D0Ch			
047D0Dh			
047D0Eh	CAN0 Mailbox 16: Time Stamp		XXXXh
047D0Fh			
047D10h	CAN0 Mailbox 17: Message Identifier	C0MB17	XXXX XXXXh
047D11h			
047D12h			
047D13h			
047D14h			
047D15h	CAN0 Mailbox 17: Data Length		XXh
047D16h	CAN0 Mailbox 17: Data Field		XXXX XXXX
047D17h			XXXX XXXXh
047D18h			
047D19h			
047D1Ah			
047D1Bh			
047D1Ch			
047D1Dh			
047D1Eh	CAN0 Mailbox 17: Time Stamp		XXXXh
047D1Fh			

X: Undefined
Blanks are reserved. No access is allowed.

Table 4.32 SFR List (32)

Address	Register	Symbol	Reset Value
047D20h	CAN0 Mailbox 18: Message Identifier	C0MB18	XXXX XXXXh
047D21h			
047D22h			
047D23h			
047D24h			
047D25h	CAN0 Mailbox 18: Data Length		XXh
047D26h	CAN0 Mailbox 18: Data Field		XXXX XXXX
047D27h			XXXX XXXXh
047D28h			
047D29h			
047D2Ah			
047D2Bh			
047D2Ch			
047D2Dh			
047D2Eh	CAN0 Mailbox 18: Time Stamp		XXXXh
047D2Fh			
047D30h	CAN0 Mailbox 19: Message Identifier	C0MB19	XXXX XXXXh
047D31h			
047D32h			
047D33h			
047D34h			
047D35h	CAN0 Mailbox 19: Data Length		XXh
047D36h	CAN0 Mailbox 19: Data Field		XXXX XXXX
047D37h			XXXX XXXXh
047D38h			
047D39h			
047D3Ah			
047D3Bh			
047D3Ch			
047D3Dh			
047D3Eh	CAN0 Mailbox 19: Time Stamp		XXXXh
047D3Fh			
047D40h	CAN0 Mailbox 20: Message Identifier	C0MB20	XXXX XXXXh
047D41h			
047D42h			
047D43h			
047D44h			
047D45h	CAN0 Mailbox 20: Data Length		XXh
047D46h	CAN0 Mailbox 20: Data Field		XXXX XXXX
047D47h			XXXX XXXXh
047D48h			
047D49h			
047D4Ah			
047D4Bh			
047D4Ch			
047D4Dh			
047D4Eh	CAN0 Mailbox 20: Time Stamp		XXXXh
047D4Fh			

X: Undefined
Blanks are reserved. No access is allowed.

Table 4.33 SFR List (33)

Address	Register	Symbol	Reset Value
047D50h	CAN0 Mailbox 21: Message Identifier	C0MB21	XXXX XXXXh
047D51h			
047D52h			
047D53h			
047D54h			
047D55h	CAN0 Mailbox 21: Data Length		XXh
047D56h	CAN0 Mailbox 21: Data Field		XXXX XXXX
047D57h			XXXX XXXXh
047D58h			
047D59h			
047D5Ah			
047D5Bh			
047D5Ch			
047D5Dh			
047D5Eh	CAN0 Mailbox 21: Time Stamp		XXXXh
047D5Fh			
047D60h	CAN0 Mailbox 22: Message Identifier	C0MB22	XXXX XXXXh
047D61h			
047D62h			
047D63h			
047D64h			
047D65h	CAN0 Mailbox 22: Data Length		XXh
047D66h	CAN0 Mailbox 22: Data Field		XXXX XXXX
047D67h			XXXX XXXXh
047D68h			
047D69h			
047D6Ah			
047D6Bh			
047D6Ch			
047D6Dh			
047D6Eh	CAN0 Mailbox 22: Time Stamp		XXXXh
047D6Fh			
047D70h	CAN0 Mailbox 23: Message Identifier	C0MB23	XXXX XXXXh
047D71h			
047D72h			
047D73h			
047D74h			
047D75h	CAN0 Mailbox 23: Data Length		XXh
047D76h	CAN0 Mailbox 23: Data Field		XXXX XXXX
047D77h			XXXX XXXXh
047D78h			
047D79h			
047D7Ah			
047D7Bh			
047D7Ch			
047D7Dh			
047D7Eh	CAN0 Mailbox 23: Time Stamp		XXXXh
047D7Fh			

X: Undefined
Blanks are reserved. No access is allowed.

Table 4.34 SFR List (34)

Address	Register	Symbol	Reset Value
047D80h	CAN0 Mailbox 24: Message Identifier	C0MB24	XXXX XXXXh
047D81h			
047D82h			
047D83h			
047D84h			
047D85h	CAN0 Mailbox 24: Data Length		XXh
047D86h	CAN0 Mailbox 24: Data Field		XXXX XXXX
047D87h			XXXX XXXXh
047D88h			
047D89h			
047D8Ah			
047D8Bh			
047D8Ch			
047D8Dh			
047D8Eh	CAN0 Mailbox 24: Time Stamp		XXXXh
047D8Fh			
047D90h	CAN0 Mailbox 25: Message Identifier	C0MB25	XXXX XXXXh
047D91h			
047D92h			
047D93h			
047D94h			
047D95h	CAN0 Mailbox 25: Data Length		XXh
047D96h	CAN0 Mailbox 25: Data Field		XXXX XXXX
047D97h			XXXX XXXXh
047D98h			
047D99h			
047D9Ah			
047D9Bh			
047D9Ch			
047D9Dh			
047D9Eh	CAN0 Mailbox 25: Time Stamp		XXXXh
047D9Fh			
047DA0h	CAN0 Mailbox 26: Message Identifier	C0MB26	XXXX XXXXh
047DA1h			
047DA2h			
047DA3h			
047DA4h			
047DA5h	CAN0 Mailbox 26: Data Length		XXh
047DA6h	CAN0 Mailbox 26: Data Field		XXXX XXXX
047DA7h			XXXX XXXXh
047DA8h			
047DA9h			
047DAAh			
047DABh			
047DACh			
047DADh			
047DAEh	CAN0 Mailbox 26: Time Stamp		XXXXh
047DAFh			

X: Undefined
Blanks are reserved. No access is allowed.

Table 4.35 SFR List (35)

Address	Register	Symbol	Reset Value
047DB0h	CAN0 Mailbox 27: Message Identifier	C0MB27	XXXX XXXXh
047DB1h			
047DB2h			
047DB3h			
047DB4h			
047DB5h	CAN0 Mailbox 27: Data Length		XXh
047DB6h	CAN0 Mailbox 27: Data Field		XXXX XXXX
047DB7h			XXXX XXXXh
047DB8h			
047DB9h			
047DBAh			
047DBBh			
047DBCh			
047DBDh			
047DBEh	CAN0 Mailbox 27: Time Stamp		XXXXh
047DBFh			
047DC0h	CAN0 Mailbox 28: Message Identifier	C0MB28	XXXX XXXXh
047DC1h			
047DC2h			
047DC3h			
047DC4h			
047DC5h	CAN0 Mailbox 28: Data Length		XXh
047DC6h	CAN0 Mailbox 28: Data Field		XXXX XXXX
047DC7h			XXXX XXXXh
047DC8h			
047DC9h			
047DCAh			
047DCBh			
047DCCh			
047DCDh			
047DCEh	CAN0 Mailbox 28: Time Stamp		XXXXh
047DCFh			
047DD0h	CAN0 Mailbox 29: Message Identifier	C0MB29	XXXX XXXXh
047DD1h			
047DD2h			
047DD3h			
047DD4h			
047DD5h	CAN0 Mailbox 29: Data Length		XXh
047DD6h	CAN0 Mailbox 29: Data Field		XXXX XXXX
047DD7h			XXXX XXXXh
047DD8h			
047DD9h			
047DDAh			
047DDBh			
047DDCh			
047DDDh			
047DDEh	CAN0 Mailbox 29: Time Stamp		XXXXh
047DDFh			

X: Undefined
Blanks are reserved. No access is allowed.

Table 4.36 SFR List (36)

Address	Register	Symbol	Reset Value
047DE0h	CAN0 Mailbox 30: Message Identifier	C0MB30	XXXX XXXXh
047DE1h			
047DE2h			
047DE3h			
047DE4h			
047DE5h	CAN0 Mailbox 30: Data Length		XXh
047DE6h	CAN0 Mailbox 30: Data Field		XXXX XXXX
047DE7h			XXXX XXXXh
047DE8h			
047DE9h			
047DEAh			
047DEBh			
047DECh			
047DEDh			
047DEEh	CAN0 Mailbox 30: Time Stamp		XXXXh
047DEFh			
047DF0h	CAN0 Mailbox 31: Message Identifier	C0MB31	XXXX XXXXh
047DF1h			
047DF2h			
047DF3h			
047DF4h			
047DF5h	CAN0 Mailbox 31: Data Length		XXh
047DF6h	CAN0 Mailbox 31: Data Field		XXXX XXXX
047DF7h			XXXX XXXXh
047DF8h			
047DF9h			
047DFAh			
047DFBh			
047DFCh			
047DFDh			
047DFEh	CAN0 Mailbox 31: Time Stamp		XXXXh
047DFFh			
047E00h	CAN0 Acceptance Mask Register 0	C0MKR0	XXXX XXXXh
047E01h			
047E02h			
047E03h			
047E04h	CAN0 Acceptance Mask Register 1	C0MKR1	XXXX XXXXh
047E05h			
047E06h			
047E07h			
047E08h	CAN0 Acceptance Mask Register 2	C0MKR2	XXXX XXXXh
047E09h			
047E0Ah			
047E0Bh			
047E0Ch	CAN0 Acceptance Mask Register 3	C0MKR3	XXXX XXXXh
047E0Dh			
047E0Eh			
047E0Fh			

X: Undefined
Blanks are reserved. No access is allowed.

Table 4.37 SFR List (37)

Address	Register	Symbol	Reset Value
047E10h	CAN0 Acceptance Mask Register 4	C0MKR4	XXXX XXXXh
047E11h			
047E12h			
047E13h			
047E14h	CAN0 Acceptance Mask Register 5	C0MKR5	XXXX XXXXh
047E15h			
047E16h			
047E17h			
047E18h	CAN0 Acceptance Mask Register 6	C0MKR6	XXXX XXXXh
047E19h			
047E1Ah			
047E1Bh			
047E1Ch	CAN0 Acceptance Mask Register 7	C0MKR7	XXXX XXXXh
047E1Dh			
047E1Eh			
047E1Fh			
047E20h	CAN0 FIFO Receive ID Compare Register 0	C0FIDCR0	XXXX XXXXh
047E21h			
047E22h			
047E23h			
047E24h	CAN0 FIFO Receive ID Compare Register 1	C0FIDCR1	XXXX XXXXh
047E25h			
047E26h			
047E27h			
047E28h	CAN0 Mask Invalid Register	C0MKIVLR	XXXX XXXXh
047E29h			
047E2Ah			
047E2Bh			
047E2Ch	CAN0 Mailbox Interrupt Enable Register	C0MIER	XXXX XXXXh
047E2Dh			
047E2Eh			
047E2Fh			
047E30h			
047E31h			
047E32h			
047E33h			
047E34h			
047E35h			
047E36h			
047E37h			
047E38h			
047E39h			
047E3Ah			
047E3Bh			
047E3Ch			
047E3Dh			
047E3Eh			
047E3Fh			
047E40h to 047F1Fh			

X: Undefined
Blanks are reserved. No access is allowed.

Table 4.38 SFR List (38)

Address	Register	Symbol	Reset Value
047F20h	CAN0 Message Control Register 0	C0MCTL0	00h
047F21h	CAN0 Message Control Register 1	C0MCTL1	00h
047F22h	CAN0 Message Control Register 2	C0MCTL2	00h
047F23h	CAN0 Message Control Register 3	C0MCTL3	00h
047F24h	CAN0 Message Control Register 4	C0MCTL4	00h
047F25h	CAN0 Message Control Register 5	C0MCTL5	00h
047F26h	CAN0 Message Control Register 6	C0MCTL6	00h
047F27h	CAN0 Message Control Register 7	C0MCTL7	00h
047F28h	CAN0 Message Control Register 8	C0MCTL8	00h
047F29h	CAN0 Message Control Register 9	C0MCTL9	00h
047F2Ah	CAN0 Message Control Register 10	C0MCTL10	00h
047F2Bh	CAN0 Message Control Register 11	C0MCTL11	00h
047F2Ch	CAN0 Message Control Register 12	C0MCTL12	00h
047F2Dh	CAN0 Message Control Register 13	C0MCTL13	00h
047F2Eh	CAN0 Message Control Register 14	C0MCTL14	00h
047F2Fh	CAN0 Message Control Register 15	C0MCTL15	00h
047F30h	CAN0 Message Control Register 16	C0MCTL16	00h
047F31h	CAN0 Message Control Register 17	C0MCTL17	00h
047F32h	CAN0 Message Control Register 18	C0MCTL18	00h
047F33h	CAN0 Message Control Register 19	C0MCTL19	00h
047F34h	CAN0 Message Control Register 20	C0MCTL20	00h
047F35h	CAN0 Message Control Register 21	C0MCTL21	00h
047F36h	CAN0 Message Control Register 22	C0MCTL22	00h
047F37h	CAN0 Message Control Register 23	C0MCTL23	00h
047F38h	CAN0 Message Control Register 24	C0MCTL24	00h
047F39h	CAN0 Message Control Register 25	C0MCTL25	00h
047F3Ah	CAN0 Message Control Register 26	C0MCTL26	00h
047F3Bh	CAN0 Message Control Register 27	C0MCTL27	00h
047F3Ch	CAN0 Message Control Register 28	C0MCTL28	00h
047F3Dh	CAN0 Message Control Register 29	C0MCTL29	00h
047F3Eh	CAN0 Message Control Register 30	C0MCTL30	00h
047F3Fh	CAN0 Message Control Register 31	C0MCTL31	00h

X: Undefined
Blanks are reserved. No access is allowed.

Table 4.39 SFR List (39)

Address	Register	Symbol	Reset Value
047F40h	CAN0 Control Register	C0CTLR	0000 0101b
047F41h			0000 0000b
047F42h	CAN0 Status Register	C0STR	0000 0101b
047F43h			0000 0000b
047F44h	CAN0 Bit Configuration Register	C0BCR	00 0000h
047F45h			
047F46h			
047F47h	CAN0 Clock Select Register	C0CLKR	000X 0000b
047F48h	CAN0 Receive FIFO Control Register	C0RFCR	1000 0000b
047F49h	CAN0 Receive FIFO Pointer Control Register	C0RFPCR	XXh
047F4Ah	CAN0 Transmit FIFO Control Register	C0TFCR	1000 0000b
047F4Bh	CAN0 Transmit FIFO Pointer Control Register	C0TFPCR	XXh
047F4Ch	CAN0 Error Interrupt Enable Register	C0EIER	00h
047F4Dh	CAN0 Error Interrupt Factor Judge Register	C0EIFR	00h
047F4Eh	CAN0 Receive Error Count Register	C0RECR	00h
047F4Fh	CAN0 Transmit Error Count Register	C0TECR	00h
047F50h	CAN0 Error Code Store Register	C0ECSR	00h
047F51h	CAN0 Channel Search Support Register	C0CSSR	XXh
047F52h	CAN0 Mailbox Search Status Register	C0MSSR	1000 0000b
047F53h	CAN0 Mailbox Search Mode Register	C0MSMR	XXXX XX00b
047F54h	CAN0 Time Stamp Register	C0TSR	0000h
047F55h			
047F56h	CAN0 Acceptance Filter Support Register	C0AFSR	XXXXh
047F57h			
047F58h	CAN0 Test Control Register	C0TCR	00h
047F59h			
047F5Ah			
047F5Bh			
047F5Ch			
047F5Dh			
047F5Eh			
047F5Fh			
047F60h to 047FFFh			
048000h to 04FFFFh			

X: Undefined
Blanks are reserved. No access is allowed.

5. Resets

There are three types of operations for resetting the MCU: hardware reset, software reset, and watchdog timer reset.

5.1 Hardware Reset

A hardware reset is generated when a low signal is applied to the $\overline{\text{RESET}}$ pin under the recommended operating conditions of the supply voltage (refer to Table 5.1 for details). When the $\overline{\text{RESET}}$ pin is driven low, all pins, and oscillators are initialized, and the main clock starts oscillating. The CPU and SFRs are initialized by a low-to-high transition on the $\overline{\text{RESET}}$ pin. Then, the CPU starts executing the program from the address indicated by the reset vector. Internal RAM is not affected by a hardware reset. However, if a hardware reset occurs during a write to the internal RAM, the value is undefined.
Figure 5.1 shows an example of the reset circuit. Figure 5.2 shows the reset sequence. Table 5.1 lists pin states while the $\overline{\text{RESET}}$ pin is held low. Figure 5.3 shows CPU register states after reset. Refer to 4. “Special Function Registers (SFRs)” for details on the states of SFRs after a reset.

A. Reset when the supply voltage is stable
(1) Drive the $\overline{\text{RESET}}$ pin low.
(2) Input at least 20 clock cycles to the XIN pin.
(3) Drive the $\overline{\text{RESET}}$ pin high.

B. Reset when turning on the power
(1) Drive the $\overline{\text{RESET}}$ pin low.
(2) Raise the supply voltage to the recommended operating voltage.
(3) Wait td(P-R) ms until the internal voltage is stabilized.
(4) Input at least 20 clock cycles to the XIN pin.
(5) Drive the $\overline{\text{RESET}}$ pin high.

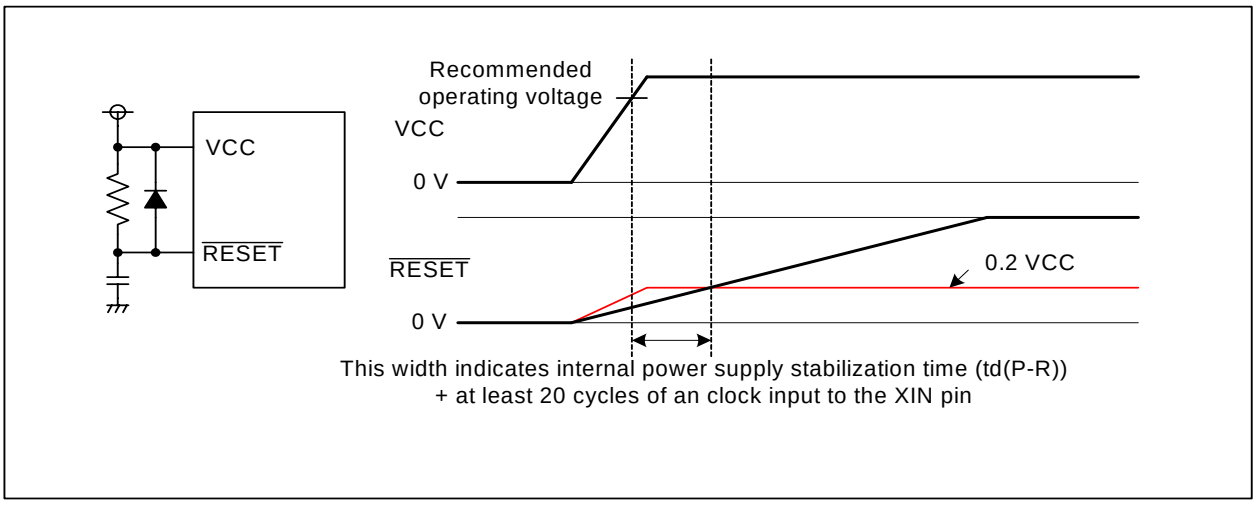
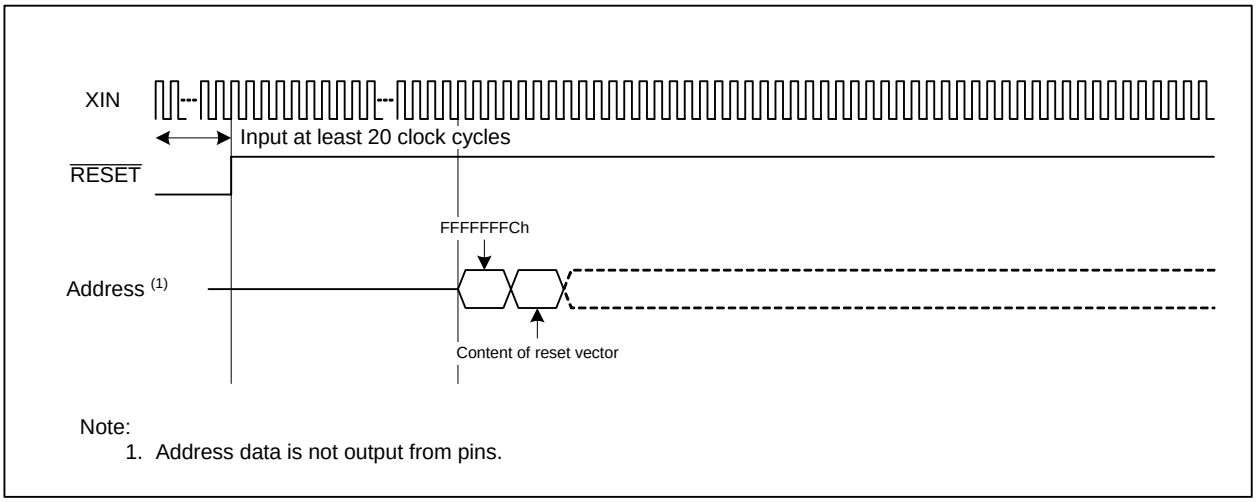


Figure 5.1 Reset Circuitry



Note:
1. Address data is not output from pins.

Figure 5.2 Reset Sequence

Table 5.1 Pin States while $\overline{\text{RESET}}$ Pin is Held Low (1)

Pin Name	Pin States
P0 to P15	Input port (high-impedance)

Note:
1. Whether a pull-up resistor is enabled or not is undefined until the internal voltage has stabilized.

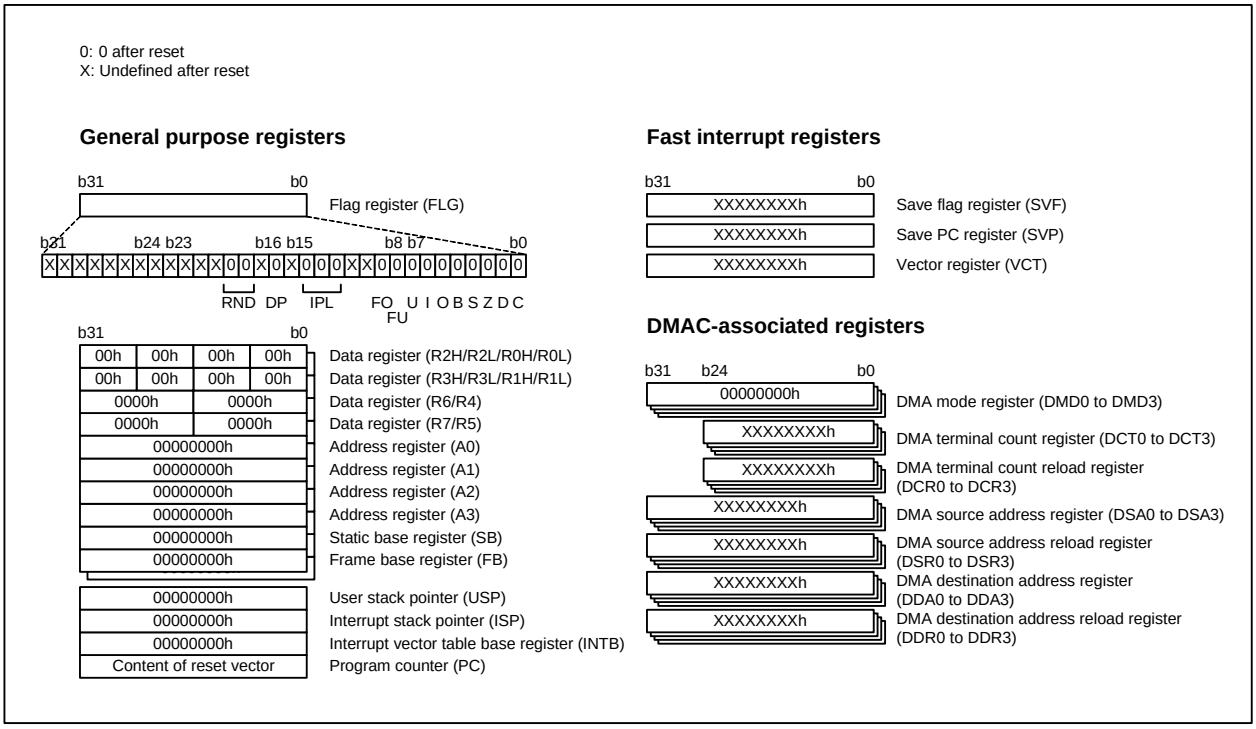

0: 0 after reset
X: Undefined after reset
General purpose registers
b31 b0
Flag register (FLG)
b31 b24 b23 b16 b15 b8 b7 b0
RND DP IPL FO U I O B S Z D C
FU
Data register (R2H/R2L/R0H/R0L)
Data register (R3H/R3L/R1H/R1L)
Data register (R6/R4)
Data register (R7/R5)
Address register (A0)
Address register (A1)
Address register (A2)
Address register (A3)
Static base register (SB)
Frame base register (FB)
User stack pointer (USP)
Interrupt stack pointer (ISP)
Interrupt vector table base register (INTB)
Program counter (PC)
Content of reset vector
Fast interrupt registers
Save flag register (SVF)
Save PC register (SVP)
Vector register (VCT)
DMAC-associated registers
DMA mode register (DMD0 to DMD3)
DMA terminal count register (DCT0 to DCT3)
DMA terminal count reload register (DCR0 to DCR3)
DMA source address register (DSA0 to DSA3)
DMA source address reload register (DSR0 to DSR3)
DMA destination address register (DDA0 to DDA3)
DMA destination address reload register (DDR0 to DDR3)

Figure 5.3 CPU Registers after Reset

5.2 Software Reset

The CPU, SFRs, and pins are initialized when the PM03 bit in the PM0 register is set to 1 (the MCU is reset). Then, the CPU executes the program from the address indicated by the reset vector. Figure 5.4 shows the PM0 register.

The PM03 bit should be set to 1 while the PLL clock is selected as the CPU clock source and the main clock oscillation is completely stable.

Processor Mode Register 0 (1)

b7	b6	b5	b4	b3	b2	b1	b0
1	0	0	0		0	0	0

Symbol	Address	Reset Value
PM0	40044h	1000 0000b

Bit Symbol	Bit Name	Function	RW
— (b2-b0)	Reserved	Should be written with 0	RW
PM03	Software Reset Bit	The MCU is reset when this bit is set to 1. The bit is read as 0	RW
— (b6-b4)	Reserved	Should be written with 0	RW
— (b7)	Reserved	Should be written with 1	RW

Note:
1. Set the PRC1 bit in the PRCR register to 1 (write enabled) before rewriting this register.

Figure 5.4 PM0 Register

5.3 Watchdog Timer Reset

The CPU, SFRs, and pins are initialized when the watchdog timer underflows while the CM06 bit in the CM0 register is 1 (reset when watchdog timer underflows). Then, the CPU executes the program from the address indicated by the reset vector.

5.4 Reset Vector

The reset vector in the R32C/100 Series is configured as shown in Figure 5.5.
The start address of a program consists of the upper 30 bits of the reset vector and 00b as lower 2 bits.
The lower 2 bits of the reset vector are bits to select the external bus width in microprocessor mode.
Therefore, the start address of a program requires 4-byte alignment so that the lower 2 bits are 00b.
In single-chip mode, set the external bus width select bits to 00b.

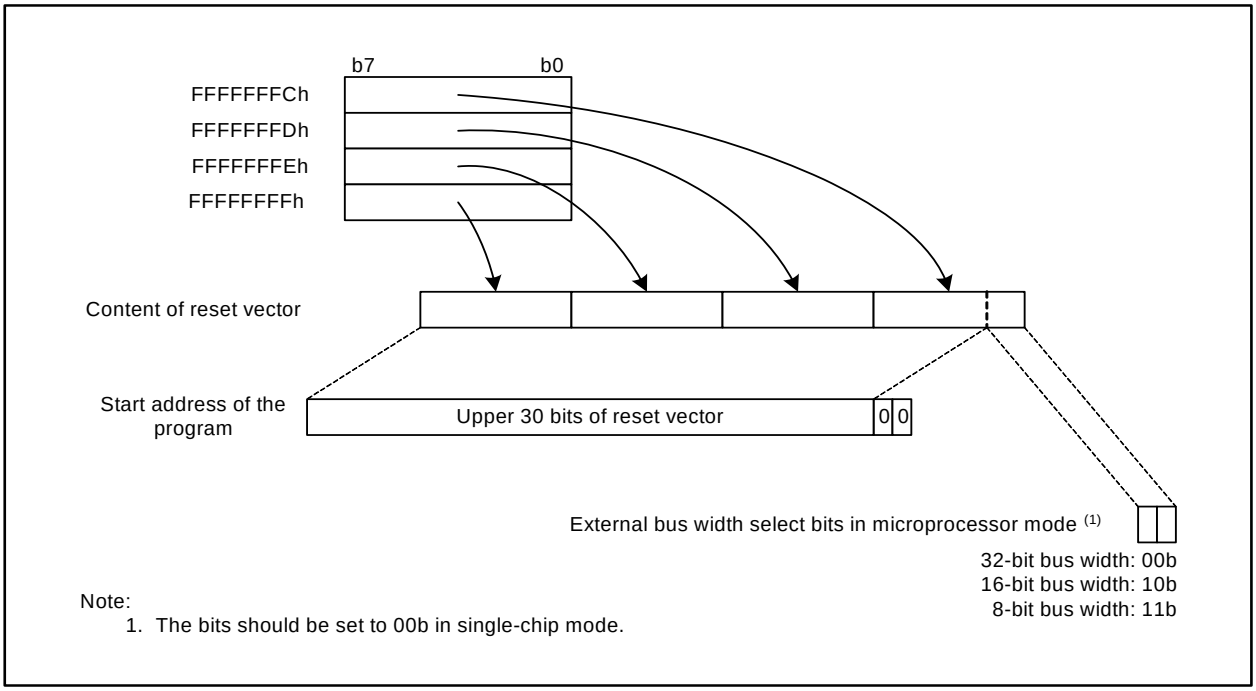


Figure 5.5 Reset Vector Configuration

6. Power Management

6.1 Voltage Regulators for Internal Logic

The supply voltage for internal logic is generated by reducing the input voltage from the VCC pin with the voltage regulators. Figure 6.1 shows a block diagram of the voltage regulators for internal logic, and Figure 6.2 shows the VRCR register.

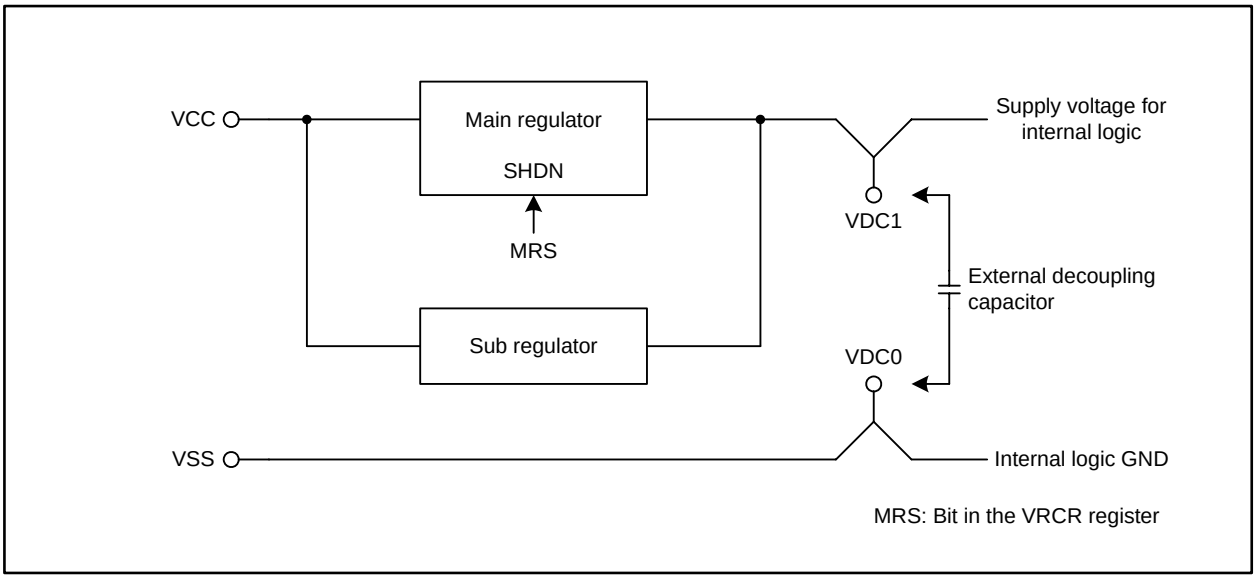


Figure 6.1 Block Diagram of Voltage Regulators for Internal Logic

Voltage Regulator Control Register (1)

b7	b6	b5	b4	b3	b2	b1	b0

Symbol	Address	Reset Value
VRCR	040060h	0000 0000b

Bit Symbol	Bit Name	Function	RW
MRS	Main Regulator Shut-down Bit (2)	0: Main regulator active 1: Main regulator stopped (3)	RW
— (b7-b1)	No register bits; should be written with 0 and read as 0		—

Notes:
1. Set the PRC31 bit in the PRCR3 register to 1 (write enabled) before rewriting this register.
2. This bit is fixed to 0 if the CM05 bit in the CM0 register is 0 (main clock oscillator enabled) or the CM10 bit in the CM1 register is 0 (PLL oscillator enabled) .
3. While the main regulator is stopped, do not rewrite the flash memory.

Figure 6.2 VRCR Register

6.1.1 Decoupling Capacitor

An external decoupling capacitor is required to stabilize internal voltage. The capacitor should be beneficially effective at higher frequencies and maintain a more stable capacitance irrespective of temperature change. In general, ceramic capacitors are recommended. The capacitance varies by conditions such as operating temperature, DC bias, and aging. To select an appropriate capacitor, these conditions should be considered. Also, refer to the recommended capacitor specifications listed in Table 6.1.

The traces between the capacitor and the VDC1/VDC0 pins should be as short and wide as physically possible.

Table 6.1 Recommended Capacitor Specifications

Temperature Characteristics				Rated Voltage (V)	Nominal Capacitance (µF)	Capacitance Tolerance (%)
Applicable standard		Operating temperature range (°C)	Capacitance change (%)			
B	JIS	-25 to 85	±10	6.3 or higher	4.7	±20 or better
R	JIS	-55 to 125	±15	6.3 or higher	4.7	±20 or better
X5R	EIA	-55 to 85	±15	6.3 or higher	4.7	±20 or better
X7R	EIA	-55 to 125	±15	6.3 or higher	4.7	±20 or better
X8R	EIA	-55 to 150	±15	6.3 or higher	4.7	±20 or better
X6S	EIA	-55 to 105	±22	6.3 or higher	4.7	±20 or better
X7S	EIA	-55 to 125	±22	6.3 or higher	4.7	±20 or better

7. Clock Generator

7.1 Clock Generator Types

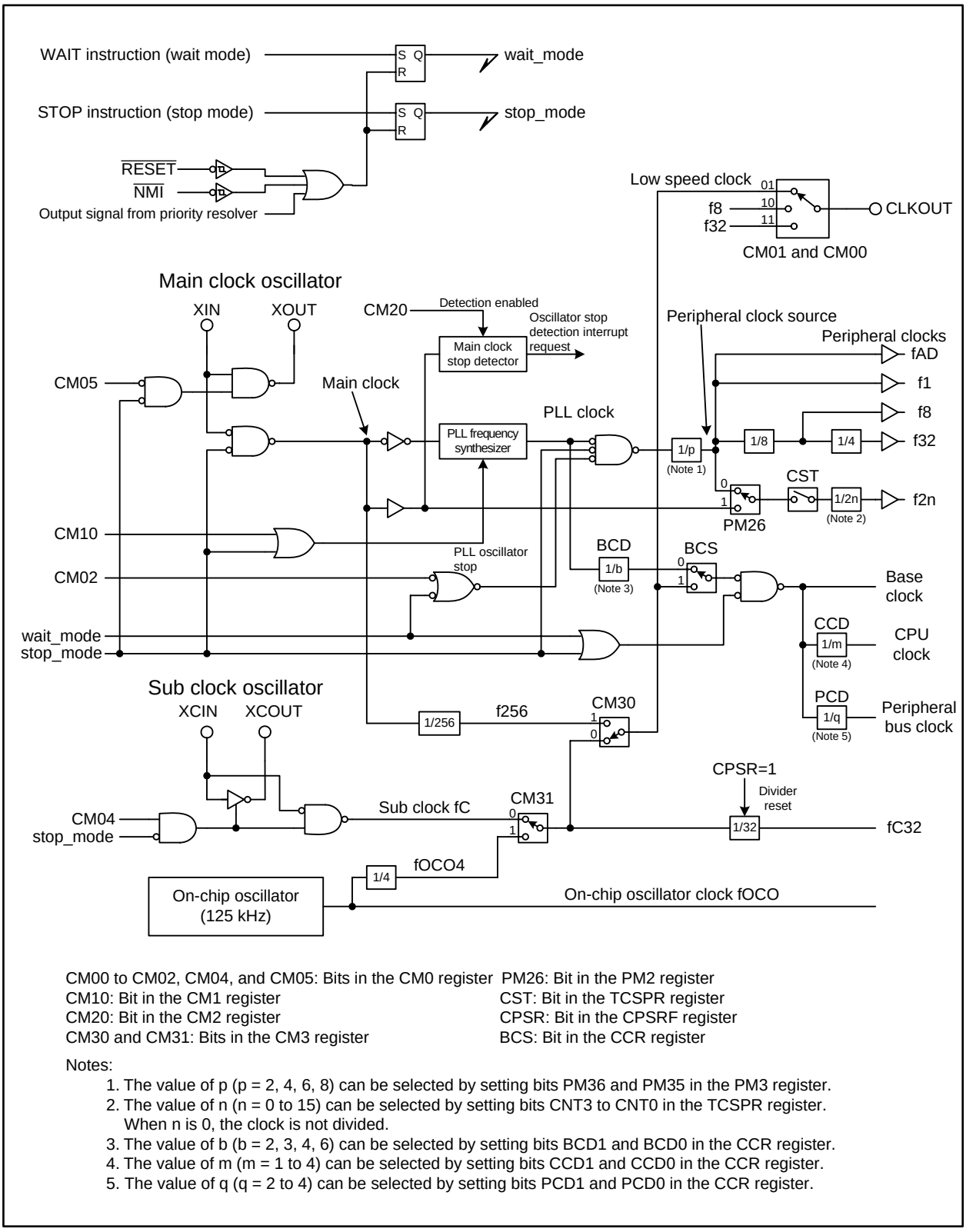
The clock generator consists of four circuits:

- Main clock oscillator
- Sub clock oscillator
- PLL frequency synthesizer
- On-chip oscillator (OCO)

Table 7.1 lists the specifications of the clock generator. Figure 7.1 shows a block diagram of the clock generator, and Figure 7.2 to Figure 7.10 show registers associated with clock control.

Table 7.1 Clock Generator Specifications

Item	Main Clock Oscillator	Sub Clock Oscillator	PLL Frequency Synthesizer	On-chip Oscillator
Used as	• PLL reference clock source • Peripheral clock source	• CPU clock source • Clock source for timers A and B	• CPU clock source • Peripheral clock source	• CPU clock source • Clock source for timers A and B
Clock frequency	4 to 16 MHz	32.768 kHz	$f_{SO(PLL)}$ or $f_{(PLL)}$	Approx. 125 kHz
Connectable oscillators or additional circuits	Ceramic resonator Crystal oscillator	Crystal oscillator	—	—
Pins for oscillators or additional circuits	XIN, XOUT	XCIN, XCOUT	—	—
Oscillator stop/ restart function	Available	Available	Available	Available
Oscillator state after a reset	Running	Stopped	Running	Stopped
Note	Externally generated clock can be input	Externally generated clock can be input	When the main clock oscillator stops running, the PLL frequency synthesizer oscillates at its own frequency of $f_{SO(PLL)}$	



CM00 to CM02, CM04, and CM05: Bits in the CM0 register
CM10: Bit in the CM1 register
CM20: Bit in the CM2 register
CM30 and CM31: Bits in the CM3 register
PM26: Bit in the PM2 register
CST: Bit in the TCSPR register
CPSR: Bit in the CPSRF register
BCS: Bit in the CCR register

Notes:
1. The value of p (p = 2, 4, 6, 8) can be selected by setting bits PM36 and PM35 in the PM3 register.
2. The value of n (n = 0 to 15) can be selected by setting bits CNT3 to CNT0 in the TCSPR register. When n is 0, the clock is not divided.
3. The value of b (b = 2, 3, 4, 6) can be selected by setting bits BCD1 and BCD0 in the CCR register.
4. The value of m (m = 1 to 4) can be selected by setting bits CCD1 and CCD0 in the CCR register.
5. The value of q (q = 2 to 4) can be selected by setting bits PCD1 and PCD0 in the CCR register.

Figure 7.1 Clock Generation Circuitry

Clock Control Register (1)

b7	b6	b5	b4	b3	b2	b1	b0
	0						

Symbol: CCR
Address: 0004h
Reset Value: 0001 1000b

Bit Symbol	Bit Name	Function	RW
BCD0	Base Clock Divide Ratio Select Bit (2)	b1 b0 0 0 : Divide-by-6 0 1 : Divide-by-4 1 0 : Divide-by-3 1 1 : Divide-by-2	RW
BCD1			RW
CCD0	CPU Clock Divide Ratio Select Bit (3)	b3 b2 0 0 : Divide-by-4 0 1 : Divide-by-3 1 0 : Divide-by-2 1 1 : No division	RW
CCD1			RW
PCD0	Peripheral Bus Clock Divide Ratio Select Bit (2, 3, 4)	b5 b4 0 0 : Do not use this combination 0 1 : Divide-by-2 1 0 : Divide-by-3 1 1 : Divide-by-4	RW
PCD1			RW
— (b6)	Reserved	Should be written with 0	RW
BCS	Base Clock Source Select Bit	0: PLL clock 1: fC, fOCO4, or f256 (5, 6)	RW

Notes:
1. Set the PRR register to AAh (write enabled) before rewriting this register.
2. The divide ratios of the base clock and peripheral bus clock should not be changed simultaneously. Doing so may cause the peripheral bus clock frequency to go over the maximum operating frequency.
3. The divide ratio of the CPU clock should be equal to or lower than that of peripheral bus clock.
4. This bit should be set only once after a reset and the setting should not be changed. To rewrite this bit, the PBC register should be rewritten first.
5. To set this bit to 1, a 32-bit write access to addresses 0004h to 0007h should be performed.
6. To use these low speed clocks, select one of them by setting bits CM31 and CM30 in the CM3 register and then set the BCS bit to 1.

Figure 7.2 CCR Register

System Clock Control Register 0 (1)

b7	b6	b5	b4	b3	b2	b1	b0
0							

Symbol	Address	Reset Value
CM0	40046h	0000 1000b

Bit Symbol	Bit Name	Function	RW
CM00	Clock Output Function Select Bit	b1 b0 0 0 : I/O port P5_3 0 1 : Output a low speed clock 1 0 : Output f8 1 1 : Output f32	RW
CM01			RW
CM02	Peripheral Clock Source Stop Bit (2)	0: Peripheral clock source not stopped in wait mode 1: Peripheral clock source stopped in wait mode (3)	RW
CM03	XCIN-XCOUT Drive Power Select Bit (4)	0: Low 1: High	RW
CM04	Port XC Switch Bit	0: I/O port 1: XCIN-XCOUT oscillator (5)	RW
CM05	Main Clock Oscillator (XIN-XOUT) Stop Bit (2, 6)	0: Main clock oscillator enabled 1: Main clock oscillator disabled	RW
CM06	Watchdog Timer Function Select Bit (7)	0: Watchdog timer interrupt 1: Reset (8)	RW
— (b7)	Reserved	Should be written with 0	RW

Notes:
1. Set the PRC0 bit in the PRCR register to 1 (write enabled) before rewriting this register.
2. When the PM21 bit in the PM2 register is 1 (clock change disabled), bits CM02 and CM05 are not changed by a write access.
3. fC32 and f2n (whose clock source is the main clock) do not stop.
4. When entering stop mode, the CM03 bit becomes 1.
5. To set the CM04 bit to 1 (XCIN-XCOUT oscillator), bits PD8_7 and PD8_6 in the PD8 register should be set to 0 (input), and the PU25 bit in the PUR2 register should be set to 0 (pull-up resistor disabled).
6. This bit stops the main clock when entering low power mode. It cannot detect whether or not the main clock oscillator stops. When this bit is set to 1, the clock applied to the XOUT pin becomes high. Since the on-chip feedback resistor remains connected, the XIN pin is connected to the XOUT pin via the feedback resistor.
7. Set this bit before activating the watchdog timer. When rewriting this bit while the watchdog timer is running, set it immediately after writing to the WDTS register.
8. Once this bit is set to 1, it cannot be set to 0 by a program.

Figure 7.3 CM0 Register

System Clock Control Register 1 (1)

b7	b6	b5	b4	b3	b2	b1	b0
0			0	0	0	0	

Symbol	Address	Reset Value
CM1	40047h	0010 0000b

Bit Symbol	Bit Name	Function	RW
CM10	PLL Oscillator Stop Bit (2, 3)	0: PLL oscillator enabled 1: PLL oscillator disabled	RW
— (b4-b1)	Reserved	Should be written with 0	RW
CM15	XIN-XOUT Drive Power Select Bit (4)	b6 b5 0 0 : Low 0 1 : High 1 0 : Super low (5) 1 1 : Do not use this combination	RW
CM16			RW
— (b7)	Reserved	Should be written with 0	RW

Notes:
1. Set the PRC0 bit in the PRCR register to 1 (write enabled) before rewriting this register.
2. When the BCS bit in the CCR register is 0 (PLL clock selected as base clock source), the CM10 bit cannot be set to 1.
3. When the PM21 bit in the PM2 register is 1 (clock change disabled), the CM10 bit cannot be changed by a write access.
4. These bits become 01b when the main clock is stopped. When setting to 00b or 10b, rewrite them after the main clock is fully stabilized.
5. The oscillator frequency should be 8 MHz or less to select super low mode.

Figure 7.4 CM1 Register

Oscillator Stop Detection Register (1)

b7	b6	b5	b4	b3	b2	b1	b0
0	0	0	0			0	

Symbol	Address	Reset Value
CM2	4004Dh	0000 0000b

Bit Symbol	Bit Name	Function	RW
CM20	Oscillator Stop Detection Enable Bit (2, 3)	0: Disable oscillator stop detection 1: Enable oscillator stop detection	RW
— (b1)	Reserved	Should be written with 0	RW
CM22	Oscillator Stop Detection Flag (4)	0: Main clock oscillator has not been stopped 1: Main clock oscillator stop detected	RW
CM23	Main Clock Monitor Flag (5)	0: Main clock oscillator active 1: Main clock oscillator stopped	RO
— (b7-b4)	Reserved	Should be written with 0	RW

Notes:
1. Set the PRC0 bit in the PRCR register to 1 (write enabled) before rewriting this register.
2. This bit should be set to 0 when f256 is selected as the base clock source in low speed mode.
3. When the PM21 bit in the PM2 register is 1 (clock change disabled), the CM02 bit cannot be changed by a write access.
4. When a main clock oscillator stop is detected, this bit becomes 1. It can be set to 0 but not to 1. If it is set to 0 while the main clock oscillator is stopped, it does not become 1 until the next main clock oscillator stop is detected.
5. The main clock state should be determined by several read accesses of this bit after an oscillator stop detection interrupt is generated.

Figure 7.5 CM2 Register

Low Speed Mode Clock Control Register (1)

b7	b6	b5	b4	b3	b2	b1	b0
X	X	X	X	X	X		

Symbol	Address	Reset Value
CM3	4005Ah	XXXX XX00b

Bit Symbol	Bit Name	Function	RW
CM30	Low Speed Mode Base Clock Select Bit	b1 b0 0 0 : fC 0 1 : f256 (main clock divided by 256) 1 0 : fOCO4 (on-chip oscillator divided by 4) (2) 1 1 : Do not use this combination	RW
CM31			RW
— (b7-b2)	No register bits; should be written with 0 and read as undefined value		—

Notes:
1. Rewrite this register after setting the PRC27 bit in the PRCR2 register to 1 (write enabled) and while the BCS bit in the CCR register is 0 (PLL clock).
2. The on-chip oscillator clock starts when the CM31 bit is set to 1.

Figure 7.6 CM3 Register

Count Source Prescaler Register

b7	b6	b5	b4	b3	b2	b1	b0
	0	0	0				

Symbol: TCSPR
Address: 035Fh
Reset Value: 0000 0000b

Bit Symbol	Bit Name	Function	RW
CNT0	Divide Ratio Select Bit (1)	f2n is either the main clock or peripheral clock source divided by 2n. If n = 0, the clock is not divided (n = setting value)	RW
CNT1			RW
CNT2			RW
CNT3			RW
— (b6-b4)	Reserved	Should be written with 0	RW
CST	Divider Operation Enable Bit	0: Stop divider operation 1: Start divider operation	RW

Note:
1. The CST bit should be set to 0 to rewrite bits CNT3 to CNT0.

Figure 7.7 TCSPR Register

Clock Prescaler Reset Register

b7	b6	b5	b4	b3	b2	b1	b0

Symbol: CPSRF
Address: 0341h
Reset Value: 0XXX XXXXb

Bit Symbol	Bit Name	Function	RW
— (b6-b0)	No register bits; should be written with 0 and read as undefined value		—
CPSR	Clock Prescaler Reset Bit	When this bit is set to 1, the fC divided-by-32 divider is initialized. The bit is read as 0	RW

Figure 7.8 CPSRF Register

Processor Mode Register 2 (1)

b7	b6	b5	b4	b3	b2	b1	b0
0		0		0	0		0

Symbol: PM2 | Address: 40053h | Reset Value: 0000 0000b

Bit Symbol	Bit Name	Function	RW
— (b0)	Reserved	Should be written with 0	RW
PM21	System Clock Protect Bit (2, 3)	0: Protect the clock by the PRCR register 1: Clock change disabled	RW
— (b3-b2)	Reserved	Should be written with 0	RW
PM24	NMI Enable Bit (2)	0: NMI disabled (4) 1: NMI enabled	RW
— (b5)	Reserved	Should be written with 0	RW
PM26	f2n Clock Source Select Bit (5)	0: Peripheral clock source 1: Main clock	RW
— (b7)	Reserved	Should be written with 0	RW

Notes:
1. Set the PRC1 bit in the PRCR register to 1 (write enabled) before rewriting this register.
2. Once this bit is set to 1, it cannot be set to 0 by a program.
3. When the PM21 bit is set to 1, the following bits are not changed by a write access:
   CM02 bit in the CM0 register (the peripheral clock source state in wait mode)
   CM05 bit in the CM0 register (main clock oscillator enabled/disabled)
   CM10 bit in the CM1 register (PLL oscillator enabled/disabled)
   CM20 bit in the CM2 register (oscillator stop detection enabled/disabled)
4. When the PM24 bit is 0, the forced cutoff of the three-phase motor control timers is also disabled.
5. Disable all the peripheral functions that use f2n before rewriting this bit.

Figure 7.9 PM2 Register

Processor Mode Register 3 (1)

b7	b6	b5	b4	b3	b2	b1	b0
0			0	0	0	0	0

Symbol: PM3
Address: 40048h
Reset Value: 0000 0000b

Bit Symbol	Bit Name	Function	RW
— (b4-b0)	Reserved	Should be written with 0	RW
PM35	Peripheral Clock Source Divide Ratio Select Bit (2)	b6 b5 0 0 : Divide-by-8 0 1 : Divide-by-6 1 0 : Divide-by-4 1 1 : Divide-by-2	RW
PM36			RW
— (b7)	Reserved	Should be written with 0	RW

Notes:
1. Set the PRC0 bit in the PRCR register to 1 (write enabled) before rewriting this register. Disable all the peripheral functions that use fAD, f1, f8, f32, or f2n (when the clock source is the peripheral clock source) to rewrite this register.
2. Select a divide ratio so that the peripheral clock source frequency does not exceed the maximum value specified in the electrical characteristics

Figure 7.10 PM3 Register

The following sections illustrate clocks generated in clock generators.

7.1.1 Main Clock

The main clock is generated by the main clock oscillator. This clock can be a clock source for the PLL reference clock or peripheral clocks. It also functions as an operating clock for the CAN module.

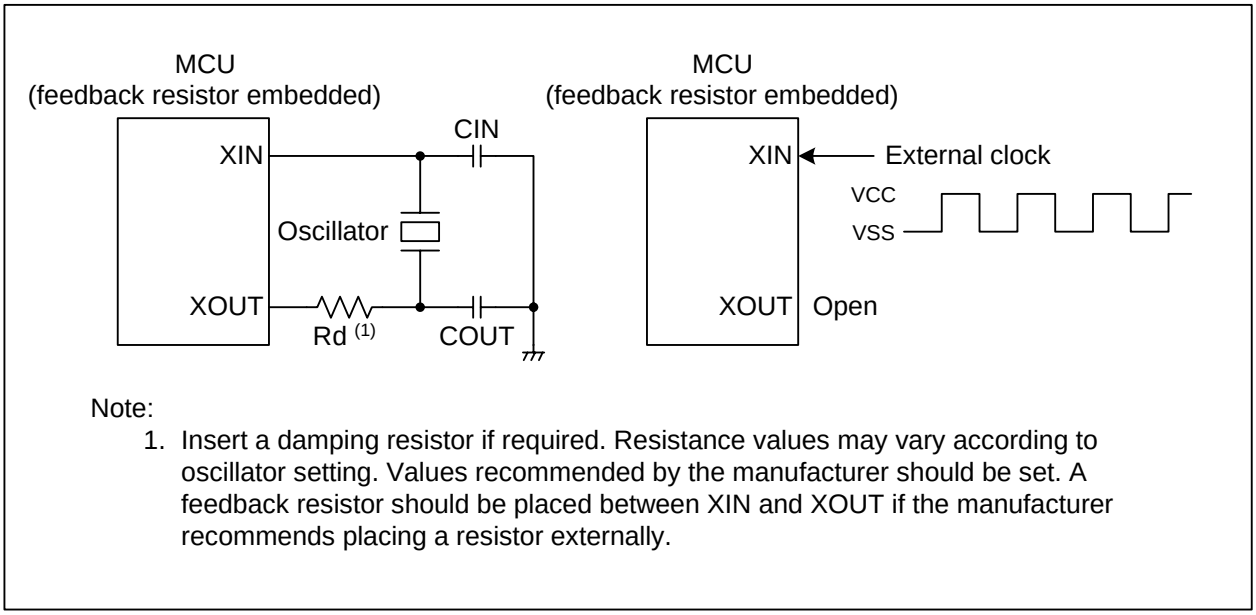
The main clock oscillator is configured with two pins, XIN and XOUT, connected by an oscillator or resonator. The circuit has an on-chip feedback resistor which is separated from the oscillator in stop mode to save power consumption. An external clock can be applied to the XIN pin in this circuit. Figure 7.11 shows an example of a main clock circuit connection.

Circuit constants vary depending on the oscillator. Circuit constants should be set as per the oscillator manufacturer's recommendations.

After a reset, the main clock oscillator is still independently active and disconnected from the PLL frequency synthesizer. A PLL frequency synthesizer self-oscillating clock which is divided by 12 is provided to the CPU.

Setting the CM05 bit in the CM0 register to 1 (main clock oscillator disabled) enables power-saving. In this case, the clock applied to the XOUT pin becomes high. The XIN pin connected to the XOUT pin by an embedded feedback resistor is also driven high. When an external clock is applied to the XIN pin, the CM05 bit should not be set to 1.

All clocks, including the main clock, stop in stop mode. Refer to 7.7 "Power Control" for details.



Note:
1. Insert a damping resistor if required. Resistance values may vary according to oscillator setting. Values recommended by the manufacturer should be set. A feedback resistor should be placed between XIN and XOUT if the manufacturer recommends placing a resistor externally.

Figure 7.11 Main Clock Circuit Connection

7.1.2 Sub Clock (fC)

The sub clock is generated by the sub clock oscillator. This clock can be a clock source for the CPU clock and a count source for timers A and B. It is output from the CLKOUT pin.

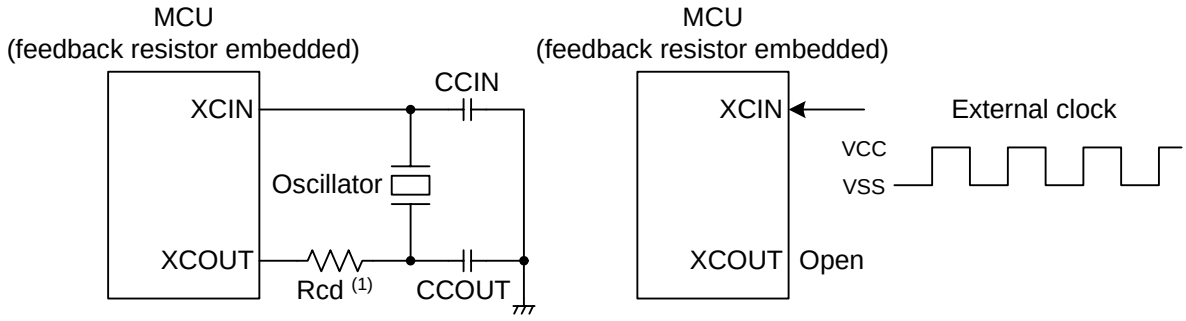
The sub clock oscillator is configured with pins XCIN and XCOUT connected by a crystal oscillator. The circuit has a on-chip feedback resistor which is separated from the oscillator in stop mode to save power consumption. An external clock can be applied to the XCIN pin. Figure 7.12 shows an example of a sub clock circuit connection. Circuit constants vary depending on the oscillator. Circuit constants should be set as per the oscillator manufacturer's recommendations.

After a reset, the sub clock oscillator is stopped. The feedback resistor is separated from the oscillator. To resume running, first set bits PD8_6 and PD8_7 in the PD8 register to 0 (input mode), and the PU25 bit in the PUR2 register to 0 (pull-up resistor unused). Then, set the CM04 bit in the CM0 register to 1 (XCIN-XCOUT oscillator).

To input an external clock to the XCIN pin, bits PD8_7 and PU25 should be set to 0, then the CM04 bit should be set to 1. The clock applied to the XCIN pin becomes a clock source for the sub clock.

When the CM3 register is set to 00h (fC selected) and the BCS bit in the CCR register is set to 1 (fC, fOCO4, or f256 is selected as base clock source) after the sub clock oscillation has stabilized, the sub clock becomes the base clock of the CPU clock and the peripheral bus clock.

All clocks, including the sub clock, stop in stop mode. Refer to 7.7 "Power Control" for details.



Note:
1. Inset a damping resistor if required. Resistance values vary according to oscillator setting. Values recommended by the manufacturer should be set. A feedback resistor should be placed between XCIN and XCOUT if the manufacturer recommends placing a resistor externally.

Figure 7.12 Sub Clock Circuit Connection

7.1.3 PLL Clock

The PLL clock is generated by the PLL frequency synthesizer based on the main clock. This clock can be a clock source for any clock including the CPU clock and the peripheral clock.
Figure 7.13 shows a block diagram of the PLL frequency synthesizer. Figure 7.14 and Figure 7.15 show registers PLC0 and PLC1, respectively.

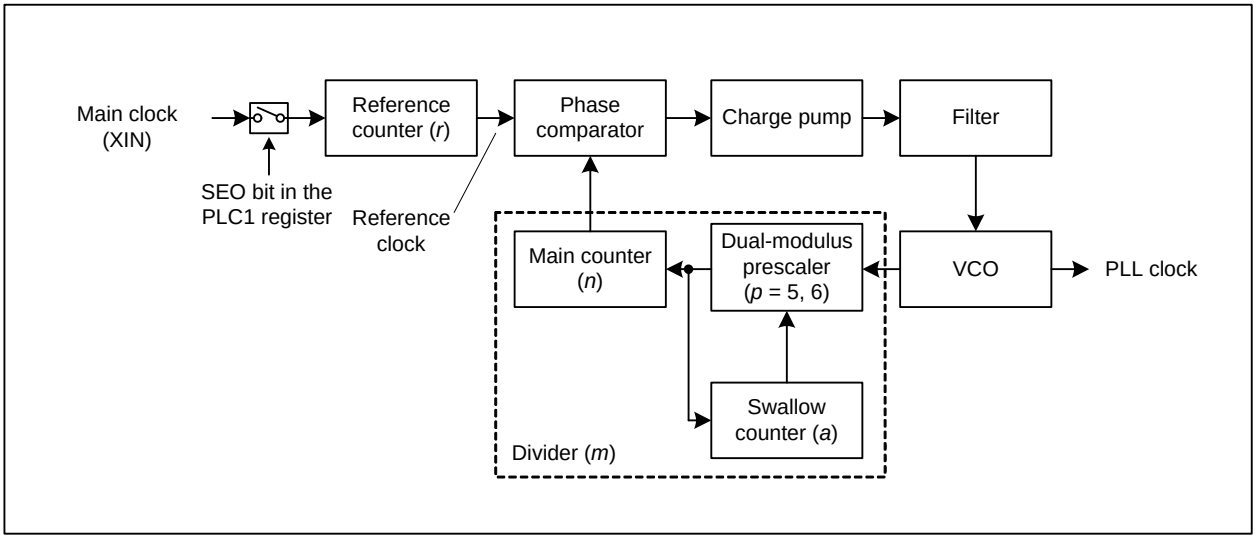


Figure 7.13 PLL Frequency Synthesizer Block Diagram

PLL Control Register 0 (1, 2)

b7	b6	b5	b4	b3	b2	b1	b0

Symbol	Address	Reset Value
PLC0	40020h	0000 0001b

Bit Symbol	Bit Name	Function	RW
MCV0	Main Counter Divide Ratio Setting Bit	Set the bits to *n* - 1 (*n* = divide ratio of the main counter)	RW
MCV1			RW
MCV2			RW
MCV3			RW
MCV4			RW
SCV0	Swallow Counter Divide Ratio Setting Bit	The divide ratio of the dual-modulus prescaler is 6 (in *a* out of *n* times) or 5 (in other cases) (*a* = setting value)	RW
SCV1			RW
SCV2			RW

Notes:
1. Set the PRC2 bit in the PRCR register to 1 (write enabled) just before rewriting this register. No interrupts or DMA transfers should be inserted between these two instructions.
2. This register can be rewritten only once after a reset.

Figure 7.14 PLC0 Register

PLL Control Register 1 (1)

b7	b6	b5	b4	b3	b2	b1	b0
0	0	0					

Symbol: PLC1 | Address: 40021h | Reset Value: 0001 1111b

Bit Symbol	Bit Name	Function	RW
RCV0	Reference Counter Divide Ratio Setting Bit	Set the bits to *r* - 1 (*r* = divide ratio of the main clock)	RW
RCV1			RW
RCV2			RW
RCV3			RW
SEO	Self-oscillating Setting Bit	0: PLL lock-in 1: Self-oscillating	RW
— (b7-b5)	Reserved	Should be written with 0	RW

Note:
1. Set the PRC2 bit in the PRCR register to 1 (write enabled) just before rewriting this register. No interrupts or DMA transfers should be inserted between these two instructions.

Figure 7.15 PLC1 Register

In the PLL frequency synthesizer, the pulse-swallow operation is implemented. The divide ratio *m* is simply expressed by *n*×*p.* However, with the swallow counter, the divide ratio *p* is 6 in *a* out of *n*, or 5 in other cases, the actual *m* is therefore given by the formula below:

$$m = n \times p$$
$$= n \times \left(\frac{a}{n} \cdot 6 + \frac{n-a}{n} \cdot 5 \right)$$
$$= 5n + a$$

The setting range of *a* is $0 \le a < 5$, $0 \le a \le n$.

As *r* is the divide ratio of the reference counter, the PLL clock has a *m/r* times the main clock (XIN) frequency.

$$\text{PLL clock frequency } f(PLL) = \frac{m}{r} \cdot \text{main clock frequency}$$
$$= \frac{5n + a}{r} \cdot \text{main clock frequency}$$

After a reset, the reference counter is divided by 16, and the PLL frequency synthesizer is multiplied by 10. Since the main clock as a reference clock is disconnected, the PLL frequency synthesizer may self-oscillate at its own frequency of $f_{SO(PLL)}$.

Each register should be set to meet the following conditions:

-The reference clock, which is the main clock divided by *r*, should be between 2 to 4 MHz.

-The divide ratio *m* is $25 \le m \le 100$.

For the setting of registers PLC1 and PLC0, Table 7.2 below should be applied. While the main clock oscillation is stable, a wait time of $t_{LOCK(PLL)}$ is necessary between rewriting registers PLC1 and PLC0, and the PLL clock becoming stable.

Table 7.2 PLC1 and PLC0 Register Settings (1)

Main Clock	*r*	Reference Clock	*n*	*a*	*m*	PLC1 Register Setting	PLC0 Register Setting	*m/r*	PLL Clock
4 MHz	2	2 MHz	9	3	48	01h	68h	24	96 MHz
6 MHz	2	3 MHz	6	2	32	01h	45h	16	96 MHz
8 MHz	3	2.6667 MHz	7	1	36	02h	26h	12	96 MHz
10 MHz	5	2 MHz	9	3	48	04h	68h	9.6	96 MHz
12 MHz	4	3 MHz	6	2	32	03h	45h	8	96 MHz
16 MHz	5	3.2 MHz	6	0	30	04h	05h	6	96 MHz
4 MHz	1	4 MHz	5	0	25	00h	04h	25	100 MHz
6 MHz	3	2 MHz	10	0	50	02h	09h	16.6667	100 MHz
8 MHz	2	4 MHz	5	0	25	01h	04h	12.5	100 MHz
10 MHz	3	3.3333 MHz	6	0	30	02h	05h	10	100 MHz
12 MHz	3	4 MHz	5	0	25	02h	04h	8.3333	100 MHz
16 MHz	4	4 MHz	5	0	25	03h	04h	6.25	100 MHz
4 MHz	1	4 MHz	6	0	30	00h	05h	30	120 MHz
6 MHz	2	3 MHz	8	0	40	01h	07h	20	120 MHz
8 MHz	2	4 MHz	6	0	30	01h	05h	15	120 MHz
10 MHz	3	3.3333 MHz	7	1	36	02h	26h	12	120 MHz
12 MHz	3	4 MHz	6	0	30	02h	05h	10	120 MHz
16 MHz	4	4 MHz	6	0	30	03h	05h	7.5	120 MHz
4 MHz	1	4 MHz	6	2	32	00h	45h	32	128 MHz
6 MHz	3	2 MHz	12	4	64	02h	8Bh	21.3333	128 MHz
8 MHz	2	4 MHz	6	2	32	01h	45h	16	128 MHz
10 MHz	5	2 MHz	12	4	64	04h	8Bh	12.8	128 MHz
12 MHz	3	4 MHz	6	2	32	02h	45h	10.6667	128 MHz
16 MHz	4	4 MHz	6	2	32	03h	45h	8	128 MHz

Note:
1. Registers PLC1 and PLC0 should be set according to the list above.

7.1.4 On-chip Oscillator Clock

The on-chip oscillator clock is generated by the on-chip oscillator (OCO). This clock can be a clock source for the CPU clock and a count source for timers A and B. This clock has a frequency of approximately 125 kHz. The on-chip oscillator clock divided by 4 can be used as the base clock for the CPU clock and peripheral bus clock.

The on-chip oscillator clock is stopped after a reset. It starts running if the CM31 bit in the CM3 register is set to 1. It is not necessary to wait for stabilization because the on-chip oscillator instantly starts oscillating.

7.2 Oscillator Stop Detection

This function detects the main clock is stopped when its oscillator stops running due to an external source.

When the CM20 bit in the CM2 register is set to 1 (oscillator stop detection enabled), an oscillator stop detection interrupt request is generated as soon as the main clock stops. Simultaneously, the PLL frequency synthesizer starts to self-oscillate at its own frequency. If the PLL frequency synthesizer is the clock source for CPU clock and peripheral clock, these clocks continue running.

When an oscillator stop is detected, the following bits in the CM2 register become 1:

- The CM22 bit: main clock oscillator stop detected
- The CM23 bit: main clock oscillator stopped
  (Refer to Figure 7.17 “State Transition When Using the Sub Clock”)

7.2.1 How to Use Oscillator Stop Detection

The oscillator stop detection interrupt shares vectors with the watchdog timer interrupt. When using these interrupts simultaneously, read the CM22 bit with an interrupt handler to determine if an oscillator stop detection interrupt request has been generated.

When the main clock oscillator resumes running after an oscillator stop is detected, the PLL clock frequency may temporarily exceed the preset value before the PLL frequency synthesizer oscillation stabilizes. As soon as an oscillator stop is detected, the main clock oscillator should be stopped from resuming (set the CM05 bit in the CM0 register to 1) or the divide ratios of the base clock and peripheral clock source should be increased by a program. The respective divide ratio can be set using bits BCD1 and BCD0 in the CCR register and bits PM36 and PM35 in the PM3 register.

In low speed mode, when the main clock oscillator stops running, an oscillator stop detection interrupt request is generated if the CM20 bit is set to 1 (oscillator stop detection enabled). The CPU clock remains running with a low speed clock source. Note that if the base clock is f256, which is the main clock divided by 256, oscillator stop detection is disabled.

Oscillator stop detection is provided to handle main clock stop caused by external sources. To stop the main clock oscillator by a program, i.e., to enter stop mode or to set the CM05 bit to 1 (main clock oscillator disabled), the CM20 bit in the CM2 register should be set to 0 (oscillator stop detection disabled). To enter wait mode, this bit should be also set to 0.

The oscillator stop detection functions depending on the voltage of a capacitor which is being changed. In more concrete terms, this function detects that the oscillator is stopped when the main clock goes lower than approximately 500 kHz. Note that if the CM22 bit is set to 0 by a program in an interrupt handler while the frequency is around 500 kHz, a stack overflow may occur due to multiple interrupt requests.

7.3 Base Clock

The base clock is a reference clock for the CPU clock and peripheral bus clock. The base clock after a reset is the PLL clock divided by 6.

The base clock source is selected between the PLL clock and the low speed clocks which contain the sub clock (fC), on-chip oscillator clock divided by 4 (fOCO4), and main clock divided by 256 (f256).

If the PLL clock is selected, it is divided by 2, 3, 4, or 6 to become the base clock. If a low speed clock is selected, the clock itself can be the base clock.

The base clock source is set using the BCS bit in the CCR register and the divide ratio for the PLL clock is set using bits BCD1 and BCD0. Bits CM31 and CM30 in the CM3 register select a low speed clock.

7.4 CPU Clock and Peripheral Bus Clock

The CPU operating clock is referred to as the CPU clock. The CPU clock after a reset is the base clock divided by 2.

The CPU clock source is the base clock and the divide ratio is selected by setting bits CCD1 and CCD0 in the CCR register. The base clock divided by 2 to 4 becomes the peripheral bus clock. Its divide ratio is selected by setting bits PCD1 and PCD0 in the CCR register. The peripheral bus clock also functions as count source for the watchdog timer and operating clock for the CAN module.

When the CPU becomes out of control, to prevent the CPU clock whose clock source is the PLL clock from stopping, the CM05 bit in the CM0 register should be set to 0 (main clock oscillator enabled) and the BCS bit in the CCR register should be set to 0 (PLL clock selected as base clock source). Then the following should be set.

(1) Set the PRC1 bit in the PRCR register to 1 (write enabled to the PM2 register).
(2) Set the PM21 bit in the PM2 register to 1 (clock change disabled).

7.5 Peripheral Clock

The peripheral clock is an operating clock or a count source for peripheral functions excluding the watchdog timer and the CAN module. The source of this clock is generated by a clock, which has the same frequency as the PLL clock, divided by 2, 4, 6, or 8 according to the settings of bits PM36 and PM35 in the PM3 register. The peripheral clock is classified into three types of clock as follows:

(1) f1, f8, f32, f2n

f1, f8, and f32 are the peripheral clock sources divided by 1, 8, and 32, respectively. The clock source for f2n is selected between the peripheral clock source and the main clock by setting the PM26 bit in the PM2 register. The f2n divide ratio can be set using bits CNT3 to CNT0 in the TCSPR register. (n = 1 to 15, not divided when n = 0)

f1, f8, f32, and f2n, whose clock source is the peripheral clock source, stop in low power mode or when the CM02 bit is set to 1 (peripheral clock source stopped in wait mode) to enter wait mode.

f1, f8, and f2n are used as a count source for timers A and B or an operating clock for the serial interface. f1 is used as an operating clock for the intelligent I/O as well.

f8 and f32 can be output from the CLKOUT pin. Refer to 7.6 “Clock Output Function” for details.

(2) fAD

fAD, which has the same frequency as peripheral clock source, is an operating clock for the A/D converter.

This clock stops in low power mode or when the CM02 bit is set to 1 (peripheral clock source stopped in wait mode) to enter wait mode.

(3) fC32

fC32, which is a sub clock divided by 32, or on-chip oscillator clock divided by 128, is used as the count source for timers A and B. This clock is available when the sub clock or on-chip oscillator clock is active.

7.6 Clock Output Function

Low speed clocks, f8, and f32 can be output from the CLKOUT pin.
Table 7.3 lists the CLKOUT pin functions.

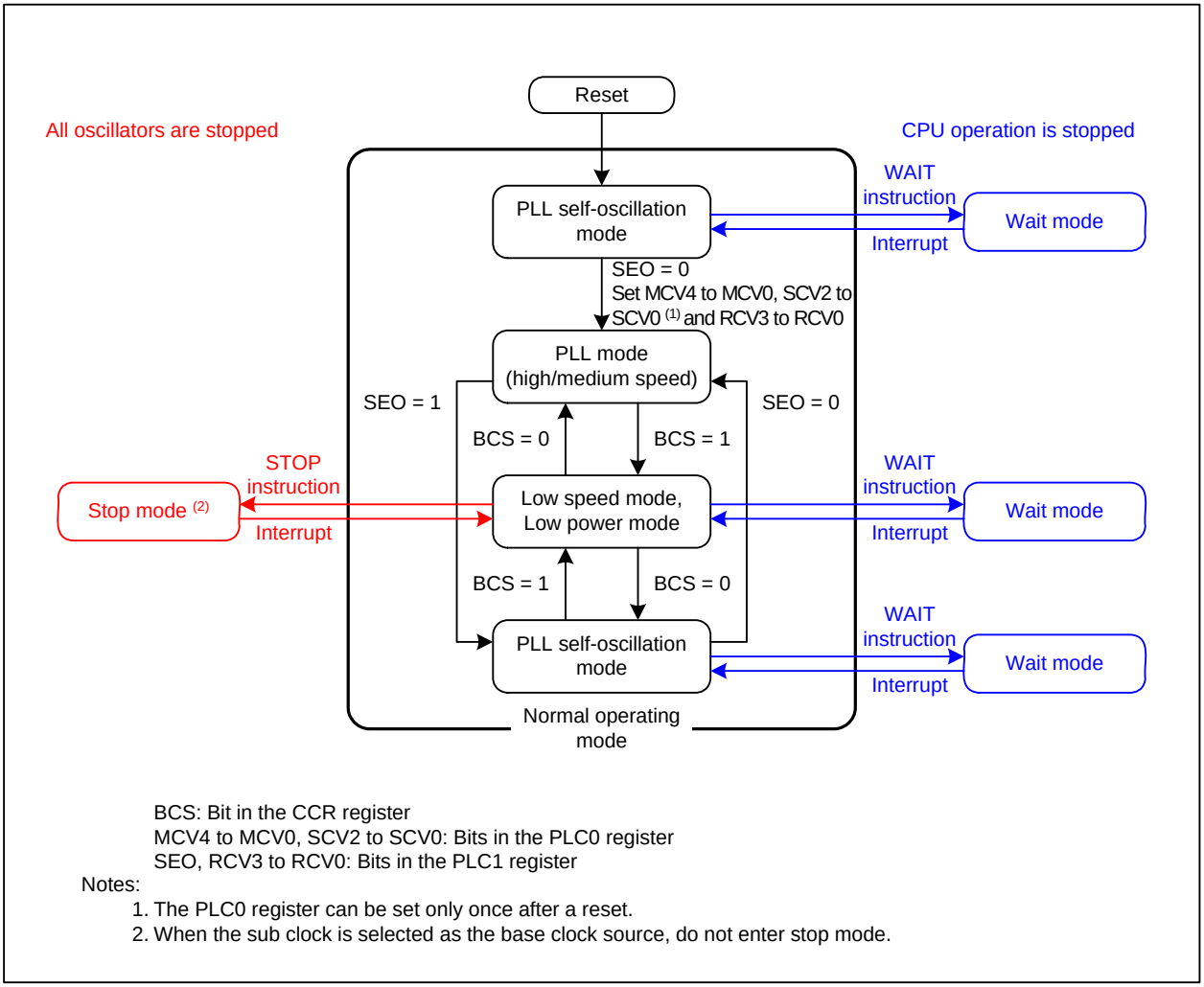
Table 7.3 CLKOUT Pin Functions

CM0 Register (1)		CLKOUT Pin Function
CM01	CM00	
0	0	I/O port P5_3
0	1	Output a low speed clock
1	0	Output f8
1	1	Output f32

Note:
1. This register should be rewritten after the PRC0 bit in the PRCR register is set to 1 (write enabled).

7.7 Power Control

Power control has three modes: wait mode, stop mode, and normal operating mode.
The name “normal operating mode” is used restrictively in this chapter, and it indicates all other modes except wait mode and stop mode. Figure 7.16 shows a block diagram of the state transition in normal operating mode, stop mode, and wait mode.



BCS: Bit in the CCR register
MCV4 to MCV0, SCV2 to SCV0: Bits in the PLC0 register
SEO, RCV3 to RCV0: Bits in the PLC1 register

Notes:
1. The PLC0 register can be set only once after a reset.
2. When the sub clock is selected as the base clock source, do not enter stop mode.

Figure 7.16 State Transition in Stop Mode and Wait Mode

7.7.1 Normal Operating Mode

Normal operating mode is classified into the five modes shown below.
In normal operating mode, the CPU clock and peripheral clock are provided to operate the CPU and peripheral functions. Power consumption is controlled by the CPU clock frequency. The higher the CPU clock frequency is, the more processing power increases. The lower the CPU clock frequency is, the less power consumption is required. Power consumption can be reduced by stopping oscillators that are not being used.

(1) PLL Mode (high speed mode)

In this mode, the PLL clock is selected as the base clock source, and the main clock is provided as the reference clock source for the PLL frequency synthesizer. High speed mode enables the CPU to operate at the maximum operating frequency. The PLL clock divided by 2 becomes the base clock. The base clock frequency should be identical to that of the CPU clock. fAD, f1, f8, f32, and f2n can be used as the peripheral clocks. When the sub clock or the on-chip oscillator clock is provided, fC32 can be used as the count source for timers A and B.

(2) PLL Mode (medium speed mode)

This mode indicates all modes in PLL mode except high speed mode. The PLL clock divided by 2, 3, 4, or 6 becomes the base clock and the base clock divided by 1 to 4 becomes the CPU clock. fAD, f1, f8, f32, and f2n can be used as the peripheral clocks. When the sub clock or the on-chip oscillator clock is provided, fC32 can be used as the count source for timers A and B.

(3) Low Speed Mode

In this mode, a low speed clock is used as the base clock source. The low speed clock becomes the base clock and the base clock divided by 1 to 4 becomes the CPU clock. fAD, f1, f8, f32, and f2n can be used as the peripheral clocks. When the sub clock or the on-chip oscillator clock is provided, fC32 can be used as the count source for timers A and B.

(4) Low Power Mode

This is a state where the main clock oscillator and the PLL frequency synthesizer are stopped after switching to low speed mode. The sub clock or the on-chip oscillator clock divided by 4 becomes the base clock and the base clock divided by 1 to 4 becomes the CPU clock. fC32, which is the only peripheral clock available, can be used as the count source for timers A and B. By setting the MRS bit in the VRCR register to 1 (main regulator stopped), this mode consumes even less power than the modes above.

(5) PLL Self-oscillation Mode

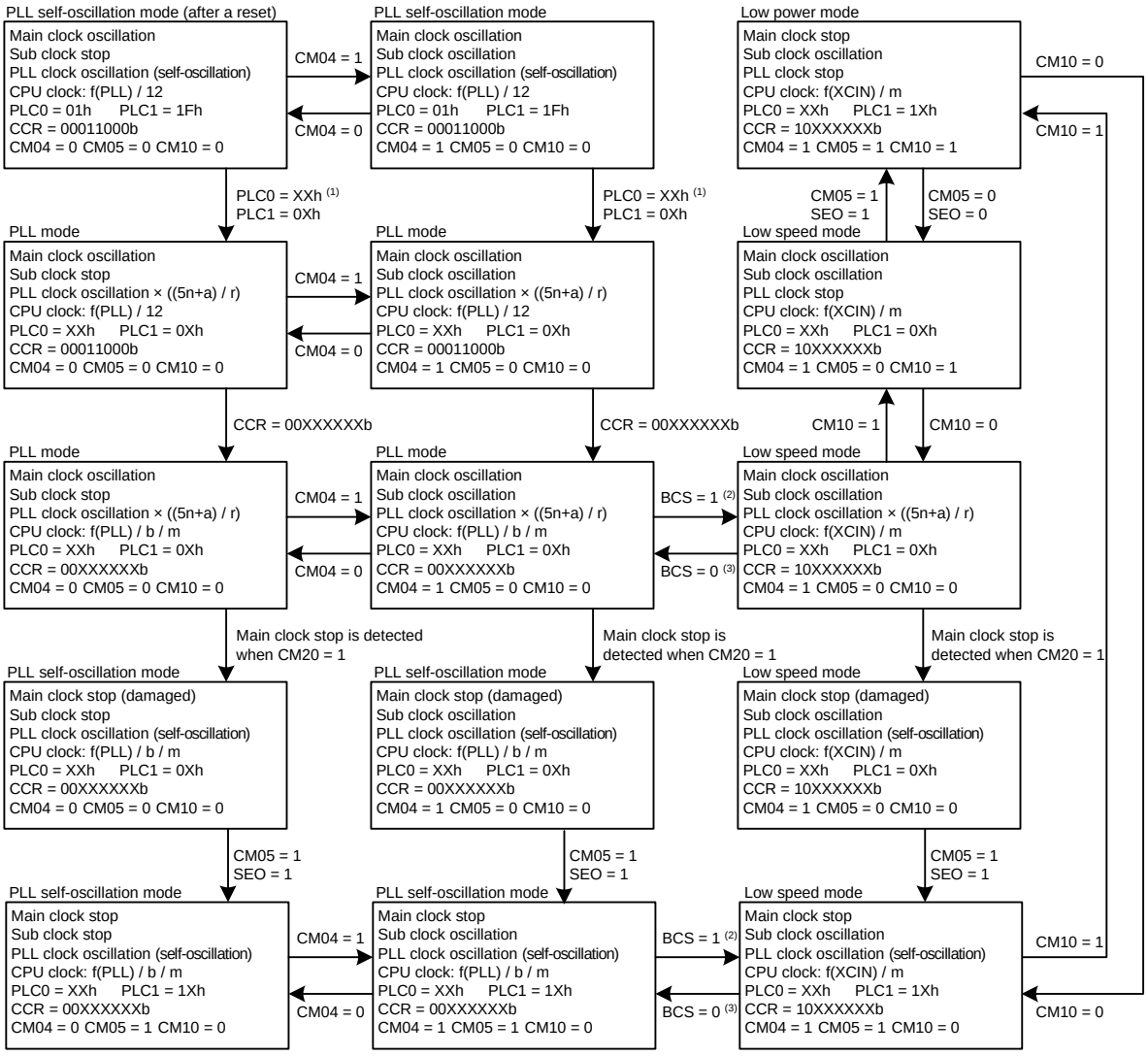
In this mode, the PLL clock is selected as the base clock source, and the main clock is not provided as the reference clock source for the PLL frequency synthesizer. The PLL frequency synthesizer self-oscillates at its own frequency. The PLL clock divided by 2, 3, 4, or 6 becomes the base clock and the base clock divided by 1 to 4 becomes the CPU clock. fAD, f1, f8, f32, and f2n can be used as the peripheral clocks. When the sub clock or the on-chip oscillator clock is provided, fC32 can be used as the count source for timers A and B.

The state transition within normal operating mode can be very complicated; therefore only the block diagrams of typical state transitions are shown. Figure 7.17 to Figure 7.19 show block diagrams of the respective state transition: state when the sub clock is used, state when the main clock divided by 256 is used, and state when the on-chip oscillator clock is used. As for the state transitions other than the above, setting of each register and the usage notes below can be used as references.

- PLL can be switched from PLL oscillating to self-oscillating by setting the SEO bit in the PLC1 register to 1. Set the SEO bit to 1 (self-oscillating) before setting the CM05 bit in the CM0 register to 0 (main clock oscillator disabled) to stop the main clock.
- The divide ratio of the clock should be increased and the frequency should be decreased by using bits BCD1 to BCD0 in the CCR register or bits PM36 to PM35 in the PM3 register before setting the SEO bit to 0 (PLL oscillating) in order to switch back PLL self-oscillation mode to PLL mode. Set back the settings of bits BCD1 to BCD0 and bits PM36 to PM35 once PLL oscillation is stabilized after setting the SEO bit to 0.
- Before switching the CPU clock to another clock, that clock should be stabilized. In particular, the sub clock oscillator may require more time to stabilize (1). Therefore, certain waiting time to switch should be taken by a program immediately after turning the MCU on or exiting stop mode.

Note:

1. Contact the oscillator manufacturer for details on oscillator stabilization time.



: Arrows indicate a one-way transition between modes. No transition should be made unless indicated.

BCS: Bit in the CCR register

CM04 and CM05: Bits in the CM0 register

CM10: Bit in the CM1 register

CM20: Bit in the CM2 register

SEO: Bit in the PLC1 register

PLC0 = XXh: The multiplication ratio of the PLL clock should be set using bits MCV4 to MCV0 and bits SCV2 to SCV0 in the PLC0 register.

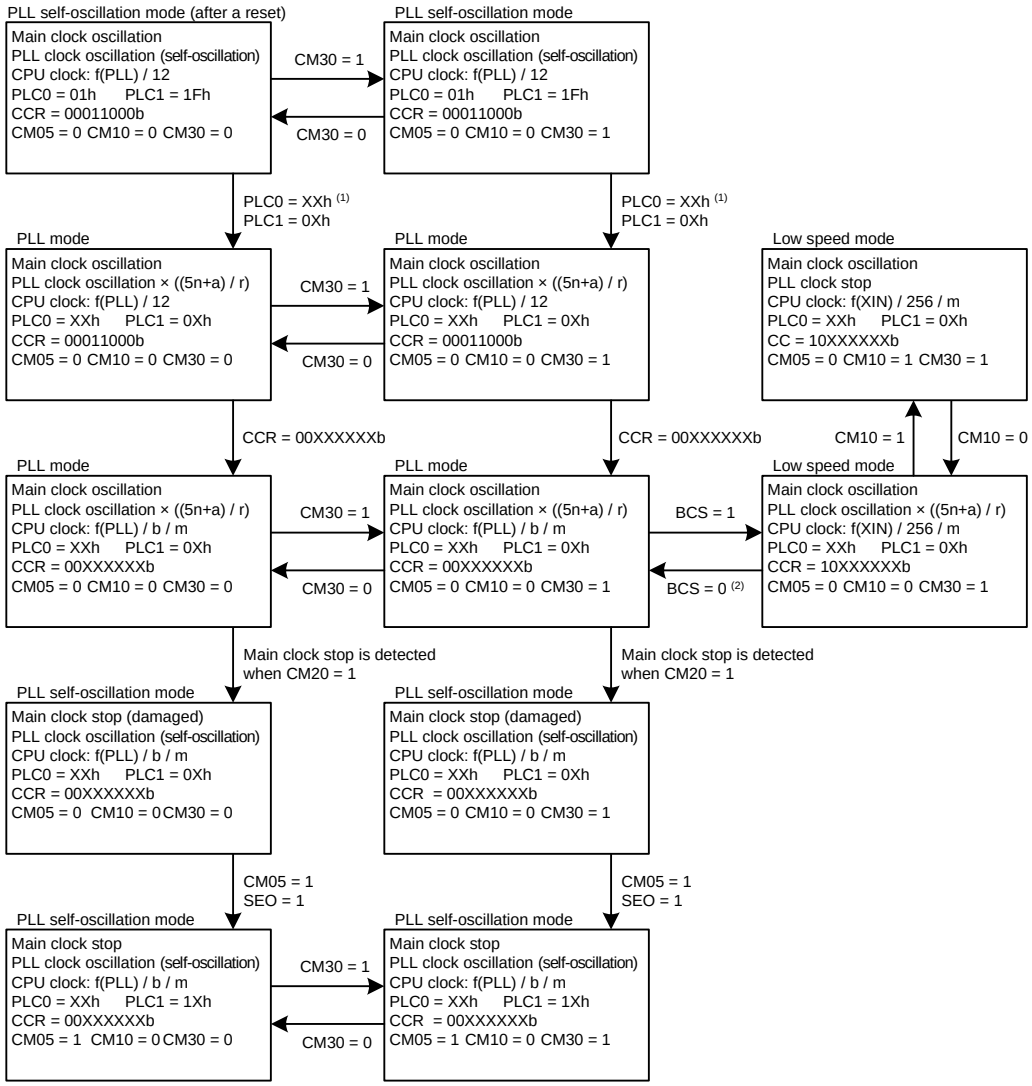
PLC1 = 0Xh: The divisor of the reference clock should be set using bits RCV3 to RCV0 in the PLC1 register. The PLL clock frequency should not exceed the maximum value specified in the electrical characteristics.

CCR = 00XXXXXXb: The divisor of each clock should be set using the CCR register. The CPU clock frequency and the peripheral bus clock frequency should not exceed the maximum values specified in the electrical characteristics.

Notes:

1. The PLC0 register can be set only once after a reset.
2. This clock should be switched after the sub clock oscillation is fully stabilized.
3. This clock should be switched after the PLL clock oscillation is fully stabilized.

Figure 7.17 State Transition When Using the Sub Clock


PLL self-oscillation mode (after a reset)
Main clock oscillation
PLL clock oscillation (self-oscillation)
CPU clock: f(PLL) / 12
PLC0 = 01h PLC1 = 1Fh
CCR = 00011000b
CM05 = 0 CM10 = 0 CM30 = 0
CM30 = 1
CM30 = 0
PLL self-oscillation mode
Main clock oscillation
PLL clock oscillation (self-oscillation)
CPU clock: f(PLL) / 12
PLC0 = 01h PLC1 = 1Fh
CCR = 00011000b
CM05 = 0 CM10 = 0 CM30 = 1
PLC0 = XXh (1)
PLC1 = 0Xh
PLL mode
Main clock oscillation
PLL clock oscillation × ((5n+a) / r)
CPU clock: f(PLL) / 12
PLC0 = XXh PLC1 = 0Xh
CCR = 00011000b
CM05 = 0 CM10 = 0 CM30 = 0
CM30 = 1
CM30 = 0
PLL mode
Main clock oscillation
PLL clock oscillation × ((5n+a) / r)
CPU clock: f(PLL) / 12
PLC0 = XXh PLC1 = 0Xh
CCR = 00011000b
CM05 = 0 CM10 = 0 CM30 = 1
Low speed mode
Main clock oscillation
PLL clock stop
CPU clock: f(XIN) / 256 / m
PLC0 = XXh PLC1 = 0Xh
CC = 10XXXXXXb
CM05 = 0 CM10 = 1 CM30 = 1
CCR = 00XXXXXXb
CM10 = 1
CM10 = 0
PLL mode
Main clock oscillation
PLL clock oscillation × ((5n+a) / r)
CPU clock: f(PLL) / b / m
PLC0 = XXh PLC1 = 0Xh
CCR = 00XXXXXXb
CM05 = 0 CM10 = 0 CM30 = 0
CM30 = 1
CM30 = 0
PLL mode
Main clock oscillation
PLL clock oscillation × ((5n+a) / r)
CPU clock: f(PLL) / b / m
PLC0 = XXh PLC1 = 0Xh
CCR = 00XXXXXXb
CM05 = 0 CM10 = 0 CM30 = 1
BCS = 1
BCS = 0 (2)
Low speed mode
Main clock oscillation
PLL clock oscillation × ((5n+a) / r)
CPU clock: f(XIN) / 256 / m
PLC0 = XXh PLC1 = 0Xh
CCR = 10XXXXXXb
CM05 = 0 CM10 = 0 CM30 = 1
Main clock stop is detected when CM20 = 1
Main clock stop is detected when CM20 = 1
PLL self-oscillation mode
Main clock stop (damaged)
PLL clock oscillation (self-oscillation)
CPU clock: f(PLL) / b / m
PLC0 = XXh PLC1 = 0Xh
CCR = 00XXXXXXb
CM05 = 0 CM10 = 0 CM30 = 0
PLL self-oscillation mode
Main clock stop (damaged)
PLL clock oscillation (self-oscillation)
CPU clock: f(PLL) / b / m
PLC0 = XXh PLC1 = 0Xh
CCR = 00XXXXXXb
CM05 = 0 CM10 = 0 CM30 = 1
CM05 = 1
SEO = 1
CM05 = 1
SEO = 1
PLL self-oscillation mode
Main clock stop
PLL clock oscillation (self-oscillation)
CPU clock: f(PLL) / b / m
PLC0 = XXh PLC1 = 1Xh
CCR = 00XXXXXXb
CM05 = 1 CM10 = 0 CM30 = 0
CM30 = 1
CM30 = 0
PLL self-oscillation mode
Main clock stop
PLL clock oscillation (self-oscillation)
CPU clock: f(PLL) / b / m
PLC0 = XXh PLC1 = 1Xh
CCR = 00XXXXXXb
CM05 = 1 CM10 = 0 CM30 = 1

→ : Arrows indicate a one-way transition between modes. No transition should be made unless indicated.

BCS: Bit in the CCR register
CM05: Bit in the CM0 register
CM10: Bit in the CM1 register
CM20: Bit in the CM2 register
CM30: Bit in the CM3 register
SEO: Bit in the PLC1 register

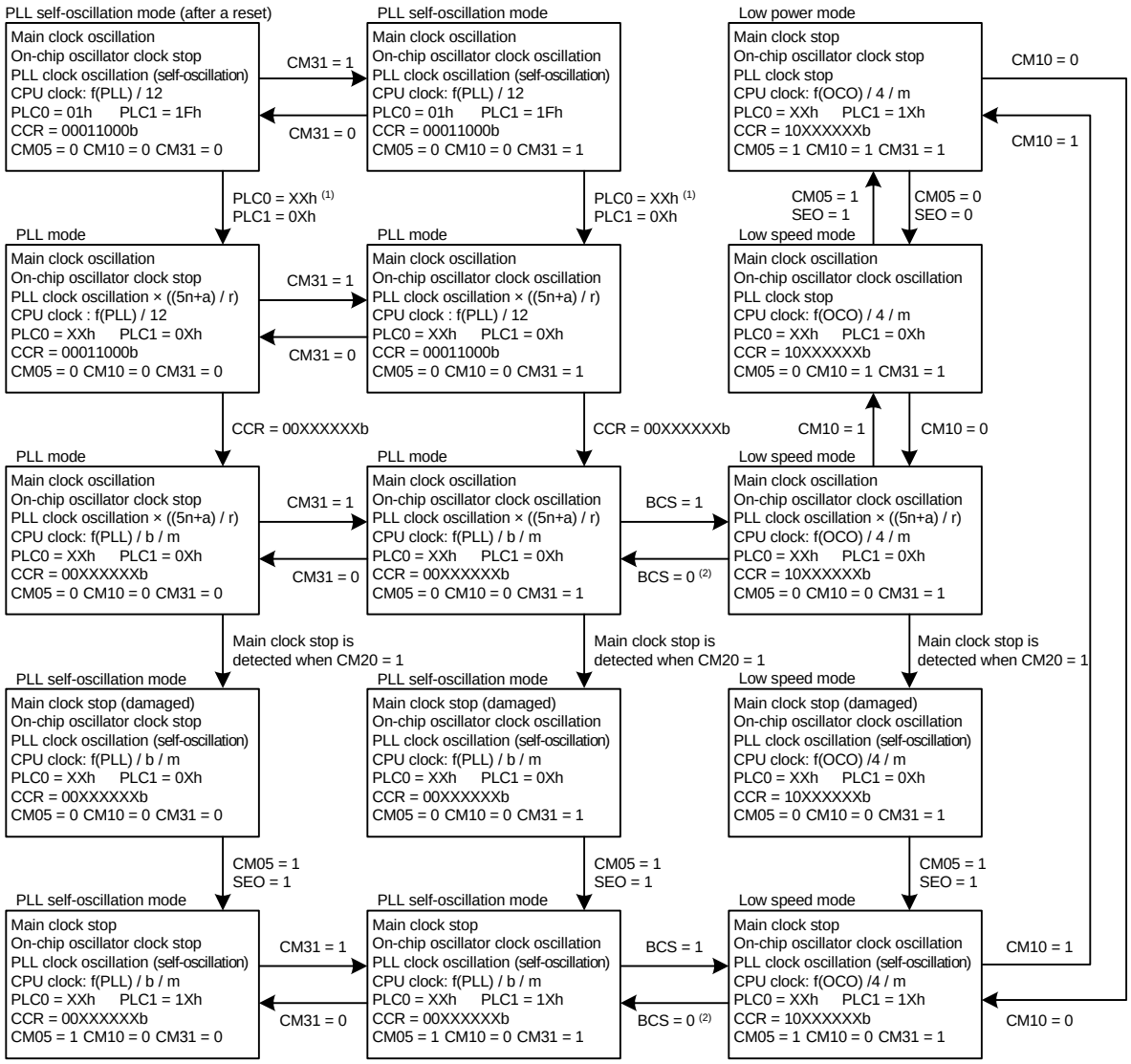
PLC0 = XXh: The multiplication ratio of the PLL clock should be set using bits MCV4 to MCV0 and bits SCV2 to SCV0 in the PLC0 register.

PLC1 = 0Xh: The divisor of the reference clock should be set using bits RCV3 to RCV0 in the PLC1 register. The PLL clock frequency should not exceed the maximum value specified in the electrical characteristics.

CCR = 00XXXXXXb: The divisor of each clock should be set using the CCR register. The CPU clock frequency and the peripheral bus clock frequency should not exceed the maximum values specified in the electrical characteristics.

Notes:
1. The PLC0 register can be set only once after a reset.
2. This clock should be switched after the PLL clock oscillation is fully stabilized.

Figure 7.18 State Transition When Using the Main Clock Divided by 256



→ : Arrows indicate a one-way transition between modes. No transition should be made unless indicated.

BCS: Bit in the CCR register
CM05: Bit in the CM0 register
CM10: Bit in the CM1 register
CM20: Bit in the CM2 register
CM31: Bit in the CM3 register
SEO: Bit in the PLC1 register

PLC0 = XXh: The multiplication ratio of the PLL clock should be set using bits MCV4 to MCV0 and bits SCV2 to SCV0 in the PLC0 register.

PLC1 = 0Xh: The divisor of the reference clock should be set using bits RCV3 to RCV0 in the PLC1 register. The PLL clock frequency should not exceed the maximum value specified in the electrical characteristics.

CCR = 00XXXXXXb: The divisor of each clock should be set using the CCR register. The CPU clock frequency and the peripheral bus clock frequency should not exceed the maximum values specified in the electrical characteristics.

Notes:
1. The PLC0 register can be set only once after a reset.
2. This clock should be switched after the PLL clock oscillation is fully stabilized.

Figure 7.19 State Transition When Using the On-chip Oscillator Clock

7.7.2 Wait Mode

As the base clock is stopped in wait mode, the CPU clock and peripheral bus clock stop running as well. The CPU and watchdog timer, operated by the CPU clock, also stop. Since the main clock, sub clock, PLL clock, and on-chip oscillator clock continue running, peripheral functions using these clocks also continue operating.

7.7.2.1 Peripheral Clock Source Stop Function

When the CM02 bit in the CM0 register is set to 1 (peripheral clock source stopped in wait mode), power consumption is reduced because peripheral clocks f1, f8, f32, f2n (when the clock source is the peripheral clock source), and fAD stop running. fC32 and f2n (when the clock source is the main clock) do not stop running.

7.7.2.2 Entering Wait Mode

To enter wait mode, the following procedures should be completed before the WAIT instruction is executed.

- Initial setting
  Set the wake-up interrupt priority level (bits RLVL2 to RLVL0 in registers RIPL1 and RIPL2) to 7. Then set each interrupt request level.
- Steps before entering wait mode
  (1) Set the I flag to 0.
  (2) Set the interrupt request level for each interrupt source (interrupt number from 1 to 127) to 0, if its interrupt request level is not 0.
  (3) Perform a dummy read of any of the interrupt control registers.
  (4) Set the processor interrupt priority level (IPL) in the flag register to 0.
  (5) Enable interrupts temporarily by executing the following instructions:
  FSET I
  NOP
  NOP
  FCLR I
  (6) Set the interrupt request level for the interrupt to exit wait mode.
  Do not rewrite the interrupt control register after this step.
  (7) Set the IPL in the flag register.
  (8) Set the interrupt priority level for resuming to the same level as the IPL.
  Interrupt request level for the interrupt to exit wait mode > IPL = Interrupt priority level for resuming
  (9) Set the CM20 bit in the CM2 register to 0 (oscillator stop detection disabled) when the oscillator stop detection is used.
  (10) Enter either PLL self-oscillation mode, low speed mode, or low power mode.
  (11) Set the I flag to 1.
  (12) Execute the WAIT instruction.
- After exiting wait mode
  Set the wake-up interrupt priority level to 7 immediately after exiting wait mode.

7.7.2.3 Pin State in Wait Mode

Table 7.4 lists the pin state in wait mode.

Table 7.4 Pin State in Wait Mode

Pin		State in Wait Mode
Ports		The state immediately before entering wait mode is held
DA0, DA1		The state immediately before entering wait mode is held
CLKOUT	When a low speed clock is selected	The clock is output
	When f8 or f32 is selected	The clock is output when the CM02 bit in the CM0 register is set to 0 (no peripheral clock source stopped in wait mode). The state immediately before entering wait mode is held when the CM02 bit is set to 1 (peripheral clock source stopped in wait mode)

7.7.2.4 Exiting Wait Mode

The MCU exits wait mode by a hardware reset, an NMI, or a peripheral interrupt assigned to software interrupt number from 0 to 63.

To exit wait mode using either a hardware reset or NMI, without using peripheral interrupts, bits ILVL2 to ILVL0 for the peripheral interrupts should be set to 000b (interrupt disabled) before executing the WAIT instruction.

The CM02 bit setting in the CM0 register affects the peripheral interrupts. When the CM02 bit is set to 0 (peripheral clock source not stopped in wait mode), peripheral interrupts for software interrupt numbers from 0 to 63 can be used to exit wait mode. When this bit is set to 1 (peripheral clock source stopped in wait mode), peripheral functions operated using clocks (f1, f8, f32, f2n whose clock source is the peripheral clock source, and fAD) generated by the peripheral clock source stop operating. Therefore, the peripheral interrupts cannot be used to exit wait mode. However, peripheral functions operated using clocks which are independent from the peripheral clock source (fC32, external clock, and f2n whose clock source is the main clock) do not stop operating. Thus, interrupts generated by peripheral functions and assigned to software interrupt numbers from 0 to 63 can be used to exit wait mode.

The CPU clock used when exiting wait mode by a peripheral interrupt or an NMI is the same clock used when the WAIT instruction is executed.

Table 7.5 lists interrupts used to exit wait mode and usage conditions.

Table 7.5 Interrupts for Exiting Wait Mode and Usage Conditions

Interrupt	When the CM02 Bit is 0	When the CM02 Bit is 1
NMI	Available	Available
External interrupt (1)	Available	Available
Key input interrupt	Available	Available
Timer A interrupt Timer B interrupt	Available in any mode	Available in event counter mode, or when the count source is fC32 or f2n (when the main clock is selected as the clock source)
Serial interface interrupt (2)	Available when an internal or external clock is used	Available when the external clock or f2n (when the main clock is selected as the clock source) is used
A/D conversion interrupt	Available in single mode or single-sweep mode	Should not be used
Intelligent I/O interrupt	Available	Should not be used
I^2C-bus interface interrupt	Available	Should not be used
I^2C-bus line interrupt	Available	Available
CAN wake-up interrupt	Available	Available

Notes:
1. INT6 to INT8 are available in the intelligent I/O interrupt only.
2. UART7 and UART8 are excluded.

7.7.3 Stop Mode

In stop mode, all of the clocks, except for those that are protected, stop running. That is, the CPU and peripheral functions, operated by the CPU clock and peripheral clock, also stop. This mode saves the most power.

7.7.3.1 Entering Stop Mode

To enter stop mode, the following procedures should be done before the STOP instruction is executed.

- Initial setting
  Set the wake-up interrupt priority level (bits RLVL2 to RLVL0 in registers RIPL1 and RIPL2) to 7. Then set each interrupt request level.
- Steps before entering stop mode
  (1) Set the I flag to 0.
  (2) Set the interrupt request level for each interrupt source (interrupt number from 1 to 127) to 0, if the interrupt request level is not 0.
  (3) Perform a dummy read of any of the interrupt control registers.
  (4) Set the processor interrupt priority level (IPL) in the flag register to 0.
  (5) Enable interrupts temporarily by executing the following instructions:
  FSET I
  NOP
  NOP
  FCLR I
  (6) Set the interrupt request level for the interrupt to exit stop mode.
  Do not rewrite the interrupt control register after this step.
  (7) Set the IPL in the flag register.
  (8) Set the interrupt priority level for resuming to the same level as the IPL.
  Interrupt request level for the interrupt to exit stop mode > IPL = Interrupt priority level for resuming
  (9) Set the CM20 bit in the CM2 register to 0 (oscillator stop detection disabled) when the oscillator stop detection is used.
  (10) Change the base clock to either the main clock divided by 256 (f256) or the on-chip oscillator clock divided by 4 (fOCO4).
  (11) Set the I flag to 1.
  (12) Execute the STOP instruction.
- After exiting stop mode
  Set the wake-up interrupt priority level to 7 immediately after exiting stop mode.

7.7.3.2 Pin State in Stop Mode

Table 7.6 lists the pin state in stop mode.

Table 7.6 Pin State in Stop Mode

Pin		State in Stop Mode
Ports		The state immediately before entering stop mode is held
DA0, DA1		The state immediately before entering stop mode is held
CLKOUT	When a low speed clock is selected	High
	When f8 or f32 is selected	The state immediately before entering stop mode is held
XIN		High-impedance
XOUT		High
XCIN, XCOUT		High-impedance

7.7.3.3 Exiting Stop Mode

The MCU exits stop mode by a hardware reset, NMI, or a peripheral interrupt assigned to software interrupt number from 0 to 63.

To exit stop mode using either a hardware reset or NMI, without using peripheral interrupts, bits ILVL2 to ILVL0 for the peripheral interrupts should be set to 000b (interrupt disabled) before executing the STOP instruction.

The CPU clock used when exiting stop mode by a peripheral interrupt or NMI is the same clock used when the STOP instruction is executed.

Table 7.7 lists interrupts used to exit stop mode and usage conditions.

Table 7.7 Interrupts for Exiting Stop Mode and Usage Conditions

Interrupt	Usage Condition
NMI	
External interrupt	INT6 to INT8 are available when intelligent I/O interrupt is used
Key input interrupt	
Timer A interrupt Timer B interrupt	Available when a timer counts an external pulse with a frequency of 100 Hz or less in event counter mode
Serial interface interrupt (1)	Available when an external clock is used
I^2C-bus line interrupt	
CAN wake-up interrupt	

Note:

1. UART7 and UART8 are excluded.

7.8 System Clock Protection

The system clock protection disables clock change when the PLL clock is selected as the base clock source. This prevents the CPU clock from stopping due to a runaway program.
When the PM21 bit in the PM2 register is set to 1 (clock change disabled), the following bits cannot be written to:

- Bits CM02 and CM05 in the CM0 register
- The CM10 bit in the CM1 register
- The CM20 bit in the CM2 register
- The PM27 bit in the PM2 register

To use the system clock protection, the CM05 bit in the CM0 register should be set to 0 (main clock oscillator enabled) and the BCS bit in the CCR register should be set to 0 (PLL clock selected as base clock source) before the following procedure is done:

(1) The PRC1 bit in the PRCR register should be set to 1 (write to the PM2 register enabled).
(2) The PM21 bit in the PM2 register should be set to 1 (clock change disabled).
(3) The PRC1 bit in the PRCR register should be set to 0 (write to the PM2 register disabled).

7.9 Notes on Clock Generator

7.9.1 Sub Clock

7.9.1.1 Oscillator Constant Matching

The constant matching of the sub clock oscillator should be evaluated in both cases when the drive power is high and low.
Contact the oscillator manufacturer for details on the oscillation circuit constant matching.

7.9.2 Power Control

Do not switch the base clock source until the oscillation of the clock to be used has stabilized. However, this does not apply to the on-chip oscillator since it starts running immediately after the CM31 bit in the CM3 register is set to 1.
To switch the base clock source from the PLL clock to a low speed clock, use the MOV.L or OR.L instruction to set the BCS bit in the CCR register to 1.

- Program example in assembly language

```
OR.L    #80h, 0004h
```

- Program example in C language

```
asm("OR.L  #80h, 0004h");
```

7.9.2.1 Stop Mode

- To exit stop mode using a reset, apply a low signal to the $\overline{\text{RESET}}$ pin until the main clock oscillation stabilizes.

7.9.2.2 Suggestions for Power Saving

The followings are suggestions to reduce power consumption when programming or designing systems.

- I/O pins:
  If inputs are floating, both transistors may be conducting. Set unassigned pins to input mode and connect each of them to VSS via a resistor, or set them to output mode and leave them open.
- A/D converter:
  When not performing the A/D conversion, set the VCUT bit in the AD0CON1 register to 0 (VREF disconnected). To perform the A/D conversion, set the VCUT bit to 1 (VREF connected) and wait 1 µs or more before starting conversion.
- D/A converter:
  When not performing the D/A conversion, set the DAiE bit in the DACON register (i = 0, 1) to 0 (output disabled) and the DAi register to 00h.
- Peripheral clock stop:
  When entering wait mode, power consumption can be reduced by setting the CM02 bit in the CM0 register to 1 to stop the peripheral clock source. However, this setting does not stop the fC32.

8. Bus

This MCU has a fast bus (CPU bus) and a slow bus (peripheral bus). Figure 8.1 shows a block diagram of the bus.

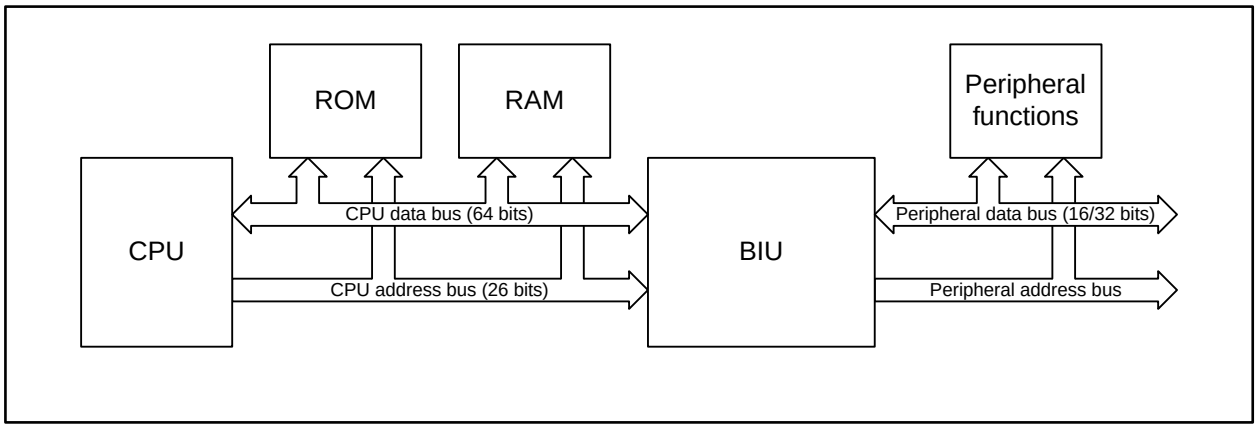


Figure 8.1 Bus Block Diagram

8.1 Bus Setting

The bus setting is controlled by the PBC register.
Table 8.1 lists the bus setting and its source.

Table 8.1 Bus Setting and Source

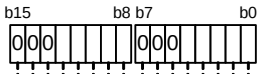
Bus Setting	Source
Internal SFR bus timing	PBC register

8.2 Peripheral Bus Timing Setting

The 16-/32-bit wide peripheral bus operates at a frequency up to 32 MHz (the theoretical value and the maximum frequency of each product group are as defined by f(BCLK) in 28. “Electrical Characteristics”). The timing adjustment and bus-width conversion with the faster, 64-bit wide CPU bus are controlled in the bus interface unit (BIU).

Figure 8.2 shows the PBC register which determines the peripheral bus timing.

Peripheral Bus Control Register (1, 2)



Symbol	Address	Reset Value
PBC	001Fh-001Eh	0504h

Bit Symbol	Bit Name	Function	RW
PRD0	Read Timing Setting Bit	Select from the three options below according to the peripheral bus clock setting (bits PCD1 and PCD0 in the CCR register). When bits PCD1 and PCD0 are set to: 1. 01b : 00100b 2. 10b : 01101b 3. 11b : 01111b	RW
PRD1			
PRD2			
PRD3			
PRD4			
— (b7-b5)	Reserved	Should be written with 0	RW
PWR0	Write Timing Setting Bit	Select from the three options below according to the peripheral bus clock setting (bits PCD1 and PCD0 in the CCR register). When bits PCD1 and PCD0 are set to: 1. 01b : 00101b 2. 10b : 01010b 3. 11b : 01111b	RW
PWR1			
PWR2			
PWR3			
PWR4			
— (b15-b13)	Reserved	Should be written with 0	RW

Notes:
1. Set the PRR register to AAh (write enabled) before rewriting this register.
2. This register should be set only once after a reset. It should not be rewritten after the CCR register is set.

Figure 8.2 PBC Register

9. Protection

This function protects important registers from being easily overwritten when a program goes out of control. It contains the following registers: PRCR, PRCR2, PRCR3, and PRR.

9.1 Protect Register (PRCR Register)

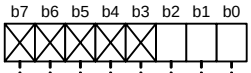
Figure 9.1 shows the PRCR register. Registers protected by the bits in the PRCR register are listed in Table 9.1.

Table 9.1 Registers Protected by the PRCR Register

Bit	Protected Registers
PRC0	CM0, CM1, CM2, and PM3
PRC1	PM0, PM2, INVC0, INVC1, IOBC, and I2CMR
PRC2	PLC0, PLC1, PD9, and P9_iS (i = 0 to 7)

The PRC2 bit becomes 0 (write disabled) when a write operation is performed in a given address after this bit is set to 1 (write enabled). In registers PD9, P9_iS (i = 0 to 7), PLC0, and PLC1, the write operation should be performed immediately after the instruction to set the PRC2 bit to 1. Any interrupt or DMA transfer should not be accepted between this instruction and the next one. Bits PRC0 and PRC1 are not set to 0 even if data is written to a given address. These bits should be set to 0 by a program.

Protect Register



Symbol: PRCR
Address: 4004Ah
Reset Value: XXXX X000b

Bit Symbol	Bit Name	Function	RW
PRC0	Protect Bit 0	Enable to write to registers CM0, CM1, CM2, and PM3 0: Write disabled 1: Write enabled	RW
PRC1	Protect Bit 1	Enable to write to registers PM0, PM2, INVC0, INVC1, IOBC, and I2CMR 0: Write disabled 1: Write enabled	RW
PRC2	Protect Bit 2 (1)	Enable to write to registers PLC0, PLC1, PD9, and P9_iS (i = 0 to 7) 0: Write disabled 1: Write enabled	RW
— (b7-b3)	No register bits; should be written with 0 and read as undefined value		—

Note:
1. The PRC2 bit becomes 0 if a write operation is performed in a given address after this bit is set to 1. Bits PRC0 and PRC1 do not automatically become 0. They should be set to 0 by a program.

Figure 9.1 PRCR Register

9.2 Protect Register 2 (PRCR2 Register)

Figure 9.2 shows the PRCR2 register which protects the CM3 register only.

Protect Register 2

b7 b6 b5 b4 b3 b2 b1 b0

Symbol	Address	Reset Value
PRCR2	4405Fh	0XXX XXXXb

Bit Symbol	Bit Name	Function	RW
— (b6-b0)	No register bits; should be written with 0 and read as undefined value		—
PRC27	CM3 Protect Bit	Enable to write to the CM3 register 0: Write disabled 1: Write enabled	RW

Figure 9.2 PRCR2 Register

9.3 Protect Register 3 (PRCR3 Register)

Figure 9.3 shows the PRCR3 register. Registers protected by the bits in the PRCR3 register are listed in Table 9.2.

Table 9.2 Registers Protected by the PRCR3 Register

Bit	Registers to be protected
PRC31	VRCR

Protect Register 3

b7 b6 b5 b4 b3 b2 b1 b0

0 0 0 0 0 0 0

Symbol	Address	Reset Value
PRCR3	4004Ch	0000 0000b

Bit Symbol	Bit Name	Function	RW
— (b0)	Reserved	Should be written with 0	RW
PRC31	Protect Bit 31	Enable to write to VRCR register 0: Write disabled 1: Write enabled	RW
— (b7-b2)	Reserved	Should be written with 0	RW

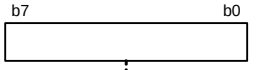
Figure 9.3 PRCR3 Register

9.4 Protect Release Register (PRR Register)

Figure 9.4 shows the PRR register. Registers protected by the PRR register are as follows: CCR, FMCR, PBC, and FEBC.

To write to the registers above, the PRR register should be set to AAh (write enabled). Otherwise, the PRR register should be set to any value other than AAh to protect the above registers from unexpected write accesses.

Protect Release Register



Symbol	Address	Reset Value
PRR	0007h	00h

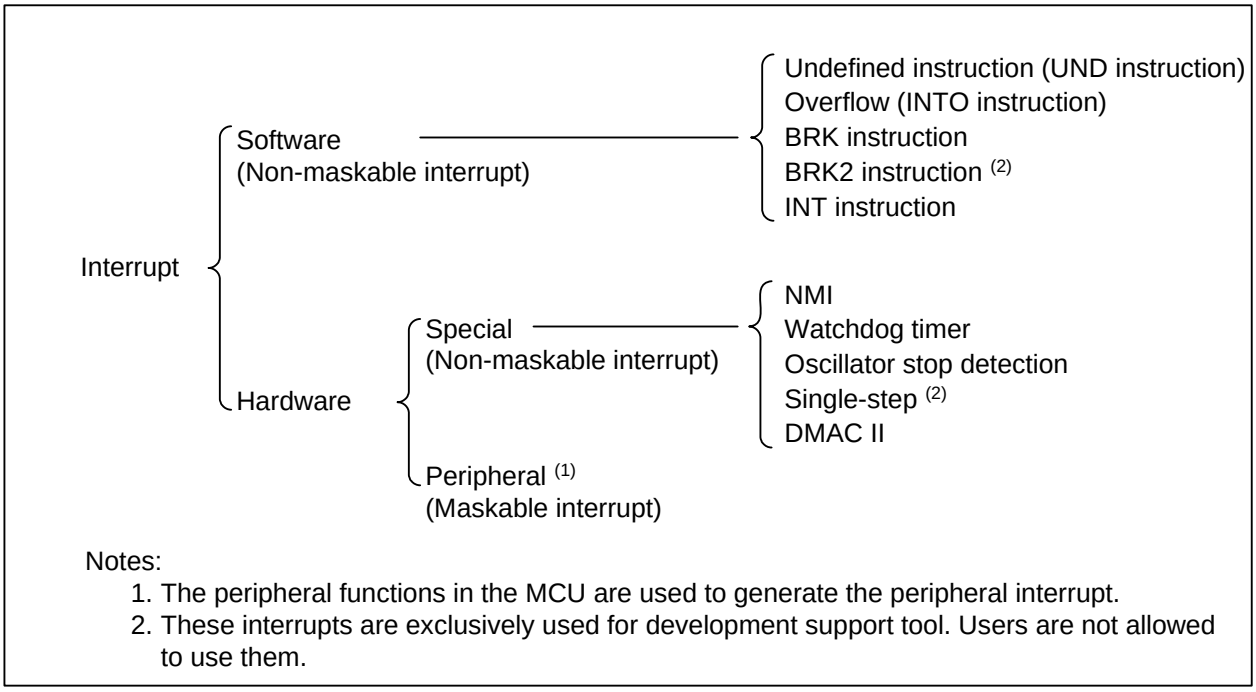
Function	Setting Range	RW
Control the protection for registers CCR, FMCR, PBC, and FEBC. AAh: Write enabled Value other than AAh: Write disabled	00h to FFh	RW

Figure 9.4 PRR Register

10. Interrupts

10.1 Interrupt Types

Figure 10.1 shows types of interrupts.



Notes:
1. The peripheral functions in the MCU are used to generate the peripheral interrupt.
2. These interrupts are exclusively used for development support tool. Users are not allowed to use them.

Figure 10.1 Interrupts

The interrupts are also classified into maskable/non-maskable.

(1) Maskable Interrupt

Maskable interrupts can be disabled by the interrupt enable flag (I flag).
The priority is configurable by assigning an interrupt request level.

(2) Non-maskable Interrupt

Maskable interrupts cannot be disabled by the interrupt enable flag (I flag).
The interrupt priority is not configurable.

10.2 Software Interrupt

Software interrupts are non-maskable. A software interrupt is generated by executing an instruction. There are five types of software interrupts as follows:

(1) Undefined Instruction Interrupt

This interrupt occurs when the UND instruction is executed.

(2) Overflow Interrupt

This interrupt occurs when the INTO instruction is executed while the O flag is 1. The following instructions may change the O flag to 1, depending on the operation result:
ABS, ADC, ADCF, ADD, ADDF, ADSF, CMP, CMPF, CNVIF, DIV, DIVF, DIVU, DIVX, EDIV, EDIVU, EDIVX, MUL, MULF, MULU, MULX, NEG, RMPA, ROUND, SBB, SCMPU, SHA, SUB, SUBF, SUNTIL, and SWHILE

(3) BRK Instruction Interrupt

This interrupt occurs when the BRK instruction is executed.

(4) BRK2 Instruction Interrupt

This interrupt occurs when the BRK2 instruction is executed.
This interrupt is only meant for use with the development support tool, and users are not allowed to use it.

(5) INT Instruction Interrupt

This interrupt occurs when the INT instruction is executed with a selected software interrupt number from 0 to 255. Numbers 0 to 127 are designated for peripheral interrupts. That is, the INT instruction with a number from 0 to 127 has the same interrupt handler as that for the peripheral interrupt.
The stack pointer (SP), which contains two types, is specified by the stack pointer select flag (U flag). For numbers 0 to 127, when an interrupt request is accepted, the U flag is saved to select the interrupt stack pointer (ISP) before the interrupt sequence is executed. The saved data of the U flag is restored upon returning from the interrupt handler. For numbers 128 to 255, the stack pointer used before the interrupt request acceptance remains unchanged for the interrupt sequence.

10.3 Hardware Interrupt

There are two kinds of hardware interrupts: special interrupt and peripheral interrupt.
In peripheral interrupts, only one interrupt with the highest priority can be specified as a fast interrupt.

10.3.1 Special Interrupt

Special interrupts are non-maskable. There are four interrupts as follows:

(1) NMI (Non Maskable Interrupt)

This interrupt occurs if an input signal at the $\overline{\text{NMI}}$ pin switches from high to low. Refer to 10.11 “NMI” for details.

(2) Watchdog Timer Interrupt

The watchdog timer generates this interrupt. Refer to 11. “Watchdog Timer” for details.

(3) Oscillator Stop Detection Interrupt

This interrupt occurs if the MCU detects a main clock oscillator stop. Refer to 7.2 “Oscillator Stop Detection” for details.

(4) Single-step Interrupt

This interrupt is only meant for use with the development support tool, and users are not allowed to use it.

10.3.2 Peripheral Interrupt

Peripheral interrupt is maskable, and is generated when an interrupt request from the peripheral functions in the MCU is accepted. It shares the interrupt vector table with software interrupt numbers 0 to 127 for the INT instruction.
Refer to Table 10.2 to Table 10.5 for details on the interrupt sources. Refer to the relevant description for details on each function.

10.4 Fast Interrupt

Fast interrupt enables the CPU to minimize the overhead of interrupt sequence. In peripheral interrupts, only one interrupt with the highest priority can be specified as the fast interrupt.

Steps to set up a fast interrupt are as follows:

(1) Set both FSIT bits in registers RIPL1 and RIPL2 to 1 (interrupt request level 7 available for fast interrupt).

(2) Set both DMAII bits in registers RIPL1 and RIPL2 to 0 (interrupt request level 7 available for interrupts).

(3) Set the start address of the fast interrupt handler to the VCT register.

Under the conditions above, bits ILVL2 to ILVL0 in the interrupt control register should be set to 111b (level 7) to enable the fast interrupt. No other interrupts should be set to interrupt request level 7.

When the fast interrupt is accepted, the flag register (FLG) and the program counter (PC) are saved to the save flag register (SVF) and the save PC register (SVP), respectively. The program is executed from the address indicated by the VCT register.

To return from the fast interrupt handler, the FREIT instruction should be executed. The values saved into the save flag register (SVF) and the save PC register (SVP) are respectively restored to the flag register (FLG) and the program counter (PC).

10.5 Interrupt Vectors

Each interrupt vector has a 4-byte memory space, in which the start address of the associated interrupt handler is stored. When an interrupt request is accepted, the instruction jumps to the address set in the interrupt vector. Figure 10.2 shows an interrupt vector.

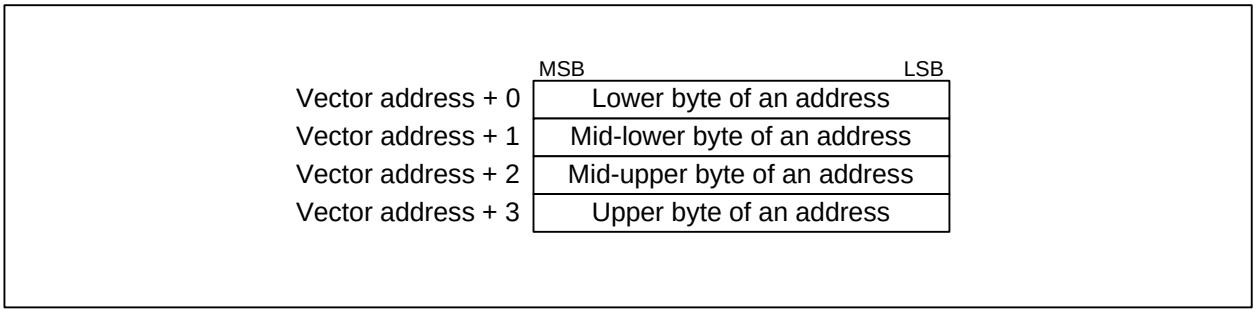


Figure 10.2 Interrupt Vector

10.5.1 Fixed Vector Table

The fixed vector table is allocated in addresses FFFFFFDCh to FFFFFFFFh. Table 10.1 lists the fixed vector table.

Table 10.1 Fixed Vector Table

Interrupt Source	Vector Table Addresses (Address (L) to Address (H))	Remarks	Reference
Undefined instruction	FFFFFFDCh to FFFFFFDFh	Interrupt by the UND instruction	R32C/100 Series Software Manual
Overflow	FFFFFFE0h to FFFFFFE3h	Interrupt by the INTO instruction	
BRK instruction	FFFFFFE4h to FFFFFFE7h	If address FFFFFFE7h is FFh, the instruction jumps to the address stored into software interrupt 0 in the relocatable vector table	
—	FFFFFFE8h to FFFFFFEBh	Reserved	
—	FFFFFFECh to FFFFFFEFh	Reserved	
Watchdog timer Oscillator stop detection	FFFFFFF0h to FFFFFFF3h	These addresses are shared by the watchdog timer interrupt and oscillator stop detection interrupt	11. “Watchdog Timer” 7. “Clock Generator”
—	FFFFFFF4h to FFFFFFF7h	Reserved	
NMI	FFFFFFF8h to FFFFFFFBh	External interrupt by the $\overline{\text{NMI}}$ pin	
Reset	FFFFFFFCh to FFFFFFFFh		5. “Resets”

10.5.2 Relocatable Vector Table

The relocatable vector table occupies a 1024-byte memory space from the start address set in the INTB register. Table 10.2 to Table 10.5. list the relocatable vector table entries.
An address in multiples of 4 should be set in the INTB register for faster interrupt sequence.

Table 10.2 Relocatable Vector Table (1/4)

Interrupt Source	Vector Table Relative Addresses (Address (L) to Address (H)) (1)	Software Interrupt Number	Reference
BRK instruction (2)	+0 to +3 (0000h to 0003h)	0	R32C/100 Series Software Manual
Reserved	+4 to +7 (0004h to 0007h)	1	
UART5 transmission, NACK (3)	+8 to +11 (0008h to 000Bh)	2	17. “Serial Interface”
UART5 reception, ACK (3)	+12 to +15 (000Ch to 000Fh)	3	
UART6 transmission, NACK (3)	+16 to +19 (0010h to 0013h)	4	
UART6 reception, ACK (3)	+20 to +23 (0014h to 0017h)	5	
Bus collision detection, start condition detection, or stop condition detection (UART5 or UART6) (3, 4)	+24 to +27 (0018h to 001Bh)	6	
Reserved	+28 to +31 (001Ch to 001Fh)	7	
DMA0 transfer complete	+32 to +35 (0020h to 0023h)	8	12. “DMAC”
DMA1 transfer complete	+36 to +39 (0024h to 0027h)	9	
DMA2 transfer complete	+40 to +43 (0028h to 002Bh)	10	
DMA3 transfer complete	+44 to +47 (002Ch to 002Fh)	11	
Timer A0	+48 to +51 (0030h to 0033h)	12	15.1 “Timer A”
Timer A1	+52 to +55 (0034h to 0037h)	13	
Timer A2	+56 to +59 (0038h to 003Bh)	14	
Timer A3	+60 to +63 (003Ch to 003Fh)	15	
Timer A4	+64 to +67 (0040h to 0043h)	16	
UART0 transmission, NACK (3)	+68 to +71 (0044h to 0047h)	17	17. “Serial Interface”
UART0 reception, ACK (3)	+72 to +75 (0048h to 004Bh)	18	
UART1 transmission, NACK (3)	+76 to +79 (004Ch to 004Fh)	19	
UART1 reception, ACK (3)	+80 to +83 (0050h to 0053h)	20	
Timer B0	+84 to +87 (0054h to 0057h)	21	15.2 “Timer B”
Timer B1	+88 to +91 (0058h to 005Bh)	22	
Timer B2	+92 to +95 (005Ch to 005Fh)	23	
Timer B3	+96 to +99 (0060h to 0063h)	24	
Timer B4	+100 to +103 (0064h to 0067h)	25	
INT5	+104 to +107 (0068h to 006Bh)	26	10.10 “External Interrupt”
INT4	+108 to +111 (006Ch to 006Fh)	27	
INT3	+112 to +115 (0070h to 0073h)	28	
INT2	+116 to +119 (0074h to 0077h)	29	
INT1	+120 to +123 (0078h to 007Bh)	30	
INT0	+124 to +127 (007Ch to 007Fh)	31	
Timer B5	+128 to +131 (0080h to 0083h)	32	15.2 “Timer B”

Notes:
1. Each entry is relative to the base address in the INTB register.
2. Interrupts from this source cannot be disabled by the I flag.
3. In I^2C mode, interrupts are generated by NACK, ACK, or detection of start condition/stop condition.
4. The IFSR16 bit in the IFSR1 register selects either the interrupt source in UART5 or that in UART6.

Table 10.3 Relocatable Vector Table (2/4)

Interrupt Source	Vector Table Relative Addresses (Address (L) to Address (H)) (1)	Software Interrupt Number	Reference
UART2 transmission, NACK (2)/I2C-bus interface (3)	+132 to +135 (0084h to 0087h)	33	17. “Serial Interface”/23. “Multi-master I2C-bus Interface”
UART2 reception, ACK (2)/I2C-bus line (3)	+136 to +139 (0088h to 008Bh)	34	
UART3 transmission, NACK (2)	+140 to +143 (008Ch to 008Fh)	35	
UART3 reception, ACK (2)	+144 to +147 (0090h to 0093h)	36	
UART4 transmission, NACK (2)	+148 to +151 (0094h to 0097h)	37	
UART4 reception, ACK (2)	+152 to +155 (0098h to 009Bh)	38	
Bus collision detection, start condition detection, or stop condition detection (UART2) (2)	+156 to +159 (009Ch to 009Fh)	39	
Bus collision detection, start condition detection, or stop condition detection (UART3 or UART0) (2, 4)	+160 to +163 (00A0h to 00A3h)	40	
Bus collision detection, start condition detection, or stop condition detection (UART4 or UART1) (2, 4)	+164 to +167 (00A4h to 00A7h)	41	
A/D0	+168 to +171 (00A8h to 00ABh)	42	18. “A/D Converter”
Key input	+172 to +175 (00ACh to 00AFh)	43	10.12 “Key Input Interrupt”
Intelligent I/O interrupt 0	+176 to +179 (00B0h to 00B3h)	44	10.13 “Intelligent I/O Interrupt”, 22. “Intelligent I/O”
Intelligent I/O interrupt 1	+180 to +183 (00B4h to 00B7h)	45	
Intelligent I/O interrupt 2	+184 to +187 (00B8h to 00BBh)	46	
Intelligent I/O interrupt 3	+188 to +191 (00BCh to 00BFh)	47	
Intelligent I/O interrupt 4	+192 to +195 (00C0h to 00C3h)	48	
Intelligent I/O interrupt 5	+196 to +199 (00C4h to 00C7h)	49	
Intelligent I/O interrupt 6	+200 to +203 (00C8h to 00CBh)	50	
Intelligent I/O interrupt 7	+204 to +207 (00CCh to 00CFh)	51	
Intelligent I/O interrupt 8	+208 to +211 (00D0h to 00D3h)	52	
Intelligent I/O interrupt 9	+212 to +215 (00D4h to 00D7h)	53	
Intelligent I/O interrupt 10	+216 to +219 (00D8h to 00DBh)	54	
Intelligent I/O interrupt 11	+220 to +223 (00DCh to 00DFh)	55	
Reserved	+224 to +227 (00E0h to 00E3h)	56	
Reserved	+228 to +231 (00E4h to 00E7h)	57	
CAN0 wakeup	+232 to +235 (00E8h to 00EBh)	58	24. “CAN Module”
Reserved	+236 to +239 (00ECh to 00EFh)	59	
Reserved	+240 to +243 (00F0h to 00F3h)	60	
Reserved	+244 to +247 (00F4h to 00F7h)	61	
Reserved	+248 to +251 (00F8h to 00FBh)	62	
Reserved	+252 to +255 (00FCh to 00FFh)	63	

Notes:

1. Each entry is relative to the base address in the INTB register.
2. In I2C mode, interrupts are generated by NACK, ACK, or detection of start condition/stop condition.
3. Select an interrupt source either of UART2 or I2C-bus interface by using the I2CEN bit in the I2CMR register.
4. The IFSR06 bit in the IFSR0 register selects either the interrupt source in UART0 or that in UART3. The IFSR07 bit selects either the interrupt source in UART1 or that in UART4.

Table 10.4 Relocatable Vector Table (3/4) (1)

Interrupt Source	Vector Table Relative Addresses (Address (L) to Address (H)) (2)	Software Interrupt Number	Reference
Reserved	+256 to +259 (0100h to 0103h)	64	
Reserved	+260 to +263 (0104h to 0107h)	65	
Reserved	+264 to +267 (0108h to 010Bh)	66	
Reserved	+268 to +271 (010Ch to 010Fh)	67	
Reserved	+272 to +275 (0110h to 0113h)	68	
Reserved	+276 to +279 (0114h to 0117h)	69	
Reserved	+280 to +283 (0118h to 011Bh)	70	
Reserved	+284 to +287 (011Ch to 011Fh)	71	
Reserved	+288 to +291 (0120h to 0123h)	72	
Reserved	+292 to +295 (0124h to 0127h)	73	
Reserved	+296 to +299 (0128h to 012Bh)	74	
Reserved	+300 to +303 (012Ch to 012Fh)	75	
Reserved	+304 to +307 (0130h to 0133h)	76	
Reserved	+308 to +311 (0134h to 0137h)	77	
Reserved	+312 to +315 (0138h to 013Bh)	78	
Reserved	+316 to +319 (013Ch to 013Fh)	79	
CAN0 transmit FIFO	+320 to +323 (0140h to 0143h)	80	24. “CAN Module”
CAN0 receive FIFO	+324 to +327 (0144h to 0147h)	81	
Reserved	+328 to +331 (0148h to 014Bh)	82	
Reserved	+332 to +335 (014Ch to 014Fh)	83	
Reserved	+336 to +339 (0150h to 0153h)	84	
Reserved	+340 to +343 (0154h to 0157h)	85	
Reserved	+344 to +347 (0158h to 015Bh)	86	
Reserved	+348 to +351 (015Ch to 015Fh)	87	
Reserved	+352 to +355 (0160h to 0163h)	88	
Reserved	+356 to +359 (0164h to 0167h)	89	
Reserved	+360 to +363 (0168h to 016Bh)	90	
Reserved	+364 to +367 (016Ch to 016Fh)	91	
Reserved	+368 to +371 (0170h to 0173h)	92	
INT8	+372 to +375 (0174h to 0177h)	93	10.10 “External Interrupt”
INT7	+376 to +379 (0178h to 017Bh)	94	
INT6	+380 to +383 (017Ch to 017Fh)	95	

Notes:
1. Entries in this table cannot be used to exit wait mode or stop mode.
2. Each entry is relative to the base address in the INTB register.

Table 10.5 Relocatable Vector Table (4/4) (1)

Interrupt Source	Vector Table Relative Addresses (Address (L) to Address (H)) (2)	Software Interrupt Number	Reference
CAN0 transmission	+384 to +387 (0180h to 0183h)	96	24. “CAN Module”
CAN0 reception	+388 to +391 (0184h to 0187h)	97	
CAN0 error	+392 to +395 (0188h to 018Bh)	98	
Reserved	+396 to +399 (018Ch to 018Fh)	99	
Reserved	+400 to +403 (0190h to 0193h)	100	
Reserved	+404 to +407 (0194h to 0197h)	101	
Reserved	+408 to +411 (0198h to 019Bh)	102	
Reserved	+412 to +415 (019Ch to 019Fh)	103	
Reserved	+416 to +419 (01A0h to 01A3h)	104	
Reserved	+420 to +423 (01A4h to 01A7h)	105	
Reserved	+424 to +427 (01A8h to 01ABh)	106	
Reserved	+428 to +431 (01ACh to 01AFh)	107	
Reserved	+432 to +435 (01B0h to 01B3h)	108	
Reserved	+436 to +439 (01B4h to 01B7h)	109	
Reserved	+440 to +443 (01B8h to 01BBh)	110	
Reserved	+444 to +447 (01BCh to 01BFh)	111	
Reserved	+448 to +451 (01C0h to 01C3h)	112	
Reserved	+452 to +455 (01C4h to 01C7h)	113	
Reserved	+456 to +459 (01C8h to 01CBh)	114	
Reserved	+460 to +463 (01CCh to 01CFh)	115	
Reserved	+464 to +467 (01D0h to 01D3h)	116	
Reserved	+468 to +471 (01D4h to 01D7h)	117	
Reserved	+472 to +475 (01D8h to 01DBh)	118	
Reserved	+476 to +479 (01DCh to 01DFh)	119	
Reserved	+480 to +483 (01E0h to 01E3h)	120	17. “Serial Interface”
Reserved	+484 to +487 (01E4h to 01E7h)	121	
Reserved	+488 to +491 (01E8h to 01EBh)	122	
Reserved	+492 to +495 (01ECh to 01EFh)	123	
UART7 transmission	+496 to +499 (01F0h to 01F3h)	124	
UART7 reception	+500 to +503 (01F4h to 01F7h)	125	
UART8 transmission	+504 to +507 (01F8h to 01FBh)	126	
UART8 reception	+508 to +511 (01FCh to 01FFh)	127	
INT instruction (3)	+0 to +3 (0000h to 0003h) to +1020 to +1023 (03FCh to 03FFh)	0 to 255	10.2 “Software Interrupt”

Notes:
1. Entries in this table cannot be used to exit wait mode or stop mode.
2. Each entry is relative to the base address in the INTB register.
3. Interrupts from this source cannot be disabled by the I flag.

10.6 Interrupt Request Acceptance

Software interrupts and special interrupts are accepted whenever their interrupt request is generated. Peripheral interrupts, however, are only accepted if the conditions below are met:

- I flag = 1
- IR bit = 1
- Bits ILVL2 to ILVL0 > IPL

The I flag, IPL, IR bit, and bits ILVL2 to ILVL0 do not affect each other. The I flag and IPL are in the flag register (FLG). The IR bit and bits ILVL2 to ILVL0 are in the interrupt control register.
The following section describes these flag and bits.

10.6.1 I Flag and IPL

The I flag (interrupt enable flag) enables or disables maskable interrupts. When the I flag is set to 1 (enabled), all maskable interrupts are enabled; when it is set to 0 (disabled), they are disabled. The I flag is automatically set to 0 after a reset.
The IPL (processor interrupt priority level), consisting of three bits, indicates eight interrupt priority levels from 0 to 7. An interrupt becomes acceptable when its interrupt request level is higher than the specified IPL (bits ILVL2 to ILVL0 > IPL).
Table 10.6 lists interrupt request levels classified by the IPL.

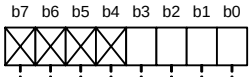
Table 10.6 Acceptable Interrupt Request Levels and IPL

Processor Interrupt Priority Level (IPL)			Acceptable Interrupt Request Levels
IPL2	IPL1	IPL0	
1	1	1	All maskable interrupts are disabled
1	1	0	Level 7 only
1	0	1	Level 6 and above
1	0	0	Level 5 and above
0	1	1	Level 4 and above
0	1	0	Level 3 and above
0	0	1	Level 2 and above
0	0	0	Level 1 and above

10.6.2 Interrupt Control Register

The interrupt control registers control each peripheral interrupt.
Figure 10.3 and Figure 10.4 show the interrupt control registers.

Interrupt Control Register



Symbol	Address	Reset Value
TA0IC to TA4IC	006Ch, 008Ch, 006Eh, 008Eh, 0070h	XXXX X000b
TB0IC to TB5IC	0094h, 0076h, 0096h, 0078h, 0098h, 0061h	XXXX X000b
S0TIC to S4TIC	0090h, 0092h, 0081h (1), 0083h, 0085h	XXXX X000b
S5TIC to S8TIC	0062h, 0064h, 00DDh, 00DFh	XXXX X000b
S0RIC to S4RIC	0072h, 0074h, 0063h (2), 0065h, 0067h	XXXX X000b
S5RIC to S8RIC	0082h, 0084h, 00FDh, 00FFh	XXXX X000b
BCN0IC to BCN3IC	0069h, 0089h, 0087h, 0069h (3)	XXXX X000b
BCN4IC to BCN6IC	0089h (4), 0066h, 0066h (5)	XXXX X000b
DM0IC to DM3IC	0068h, 0088h, 006Ah, 008Ah	XXXX X000b
AD0IC	006Bh	XXXX X000b
KUPIC	008Bh	XXXX X000b
IIO0IC to IIO5IC	006Dh, 008Dh, 006Fh, 008Fh, 0071h, 0091h	XXXX X000b
IIO6IC to IIO11IC	0073h, 0093h, 0075h, 0095h, 0077h, 0097h	XXXX X000b
I2CIC, I2CLIC	0081h (1), 0063h (2)	XXXX X000b
C0FTIC	00D0h	XXXX X000b
C0FRIC	00F0h	XXXX X000b
C0TIC	00C1h	XXXX X000b
C0RIC	00E1h	XXXX X000b
C0EIC	00C3h	XXXX X000b
C0WIC	007Bh	XXXX X000b

Bit Symbol	Bit Name	Function	RW
ILVL0	Interrupt Request Level Select Bit	b2 b1 b0 0 0 0 : Level 0 (interrupt disabled) 0 0 1 : Level 1 0 1 0 : Level 2 0 1 1 : Level 3 1 0 0 : Level 4 1 0 1 : Level 5 1 1 0 : Level 6 1 1 1 : Level 7	RW
ILVL1			RW
ILVL2			RW
IR	Interrupt Request Flag	0: No interrupt requested 1: Interrupt requested (6)	RW
— (b7-b4)	No register bits; should be written with 0 and read as undefined value		—

Notes:
1.The S2TIC register shares an address with the I2CIC register.
2.The S2RIC register shares an address with the I2CLIC register.
3.The BCN0IC register shares an address with the BCN3IC register.
4.The BCN1IC register shares an address with the BCN4IC register.
5.The BCN5IC register shares an address with the BCN6IC register.
6.This bit can be set to 0 only (It should not be set to 1).

Figure 10.3 Interrupt Control Register (1)

Interrupt Control Register

b7 b6 b5 b4 b3 b2 b1 b0

Symbol	Address	Reset Value
INT0IC to INT2IC	009Eh, 007Eh, 009Ch	XX00 X000b
INT3IC to INT5IC	007Ch, 009Ah, 007Ah	XX00 X000b
INT6IC to INT8IC	00FEh, 00DEh, 00FCh	XX00 X000b

Bit Symbol	Bit Name	Function	RW
ILVL0	Interrupt Request Level Select Bit	b2 b1 b0 0 0 0 : Level 0 (interrupt disabled) 0 0 1 : Level 1 0 1 0 : Level 2 0 1 1 : Level 3 1 0 0 : Level 4 1 0 1 : Level 5 1 1 0 : Level 6 1 1 1 : Level 7	RW
ILVL1			RW
ILVL2			RW
IR	Interrupt Request Flag	0: No interrupt requested 1: Interrupt requested (1)	RW
POL	Polarity Select Bit	0: Select the falling edge or a low 1: Select the rising edge or a high (2)	RW
LVS	Level/Edge Sensitive Select Bit	0: Edge sensitive 1: Level sensitive (3)	RW
— (b7-b6)	No register bits; should be written with 0 and read as undefined value		—

Notes:
1. This bit can be set to 0 only (it should not be set to 1).
2. This bit should be set to 0 (the falling edge or low level) to set the corresponding bit in registers IFSR0 and IFSR1 to 1 (both edges).
3. To select the level sensitive, the corresponding bit in registers IFSR0 and IFSR1 should be set to 0 (one edge).

Figure 10.4 Interrupt Control Register (2)

Bits ILVL2 to ILVL0

Bits ILVL2 to ILVL0 select the interrupt request level. The higher the level is, the higher interrupt priority is.

When an interrupt request is generated, its request level is compared to the IPL. This interrupt is accepted only when the interrupt request level is higher than the IPL. When bits ILVL2 to ILVL0 are set to 000b, the interrupt is disabled.

IR bit

The IR bit becomes 1 (interrupt requested) when an interrupt request is generated; this bit setting is retained until the interrupt request is accepted. When the request is accepted and the instruction jumps to the corresponding interrupt vector, the IR bit becomes 0 (no interrupt requested).

The IR bit can be set to 0 by a program. This bit should not be set to 1.

When rewriting the interrupt control register, no corresponding interrupt request should be generated. If it may be generated, disable all the maskable interrupts before the rewrite.
When enabling the maskable interrupts immediately after the rewrite, there should be sufficient time for the rewrite to complete before the interrupt enable flag (I flag) becomes 1. To delay the execution of the second instruction, insert NOPs or perform a dummy read of the interrupt control register after the first instruction.
If an interrupt request is generated for a register being rewritten, the IR bit may not become 1 depending on the instruction being used. If this is not desired, use one of the following instructions to rewrite the register:

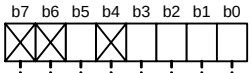
- AND
- OR
- BCLR
- BSET

When setting the IR bit to 0 by the AND or BCLR instruction, the IR bit may not become 0. This is because an interrupt request generated while the instruction above is being executed is kept pending. If this is not desired, the register should be reconfigured by the MOV instruction. To set just the IR bit to 0, first temporarily store the read value to memory or CPU-internal registers, then execute either the AND or BCLR instruction in the stored area. After that, write the value back to the register by the MOV instruction.

10.6.3 Wake-up IPL Setting Register

The wake-up IPL setting register (registers RIPL1 and RIPL2) is used for an interrupt to exit wait or stop mode, or for the fast interrupt.
Refer to 7.7.2 “Wait Mode”, 7.7.3 “Stop Mode”, or 10.4 “Fast Interrupt” for details.
Figure 10.5 shows registers RIPL1 and RIPL2.

Wake-up IPL Setting Register i (i = 1, 2) (1)



Symbol	Address	Reset Value
RIPL1, RIPL2	4407Fh, 4407Dh	XX0X 0000b

Bit Symbol	Bit Name	Function	RW
RLVL0	Interrupt Priority Level for Wake-up Select Bit (2)	b2 b1 b0 0 0 0 : Level 0 0 0 1 : Level 1 0 1 0 : Level 2 0 1 1 : Level 3 1 0 0 : Level 4 1 0 1 : Level 5 1 1 0 : Level 6 1 1 1 : Level 7	RW
RLVL1			RW
RLVL2			RW
FSIT	Fast Interrupt Select Bit (3)	0: Use interrupt request level 7 for normal interrupt 1: Use interrupt request level 7 for fast interrupt (4)	RW
— (b4)	No register bit; should be written with 0 and read as undefined value		—
DMAII	DMA II Select Bit (5)	0: Use interrupt request level 7 for interrupt 1: Use interrupt request level 7 for DMA II transfer (4)	RW
— (b7-b6)	No register bits; should be written with 0 and read as undefined value		—

Notes:
1. Registers RIPL1 and RIPL2 should be identically set.
2. The MCU exits wait mode or stop mode if the request level of requested interrupt is higher than the level selected using bits RLVL2 to RLVL0. These bits should be set to the same value as the IPL in the flag register (FLG).
3. When the FSIT bit is set to 1, an interrupt with interrupt request level 7 becomes the fast interrupt. In this case only one interrupt should be set to the interrupt request level 7.
4. Either the FSIT or DMAII bit should be set to 1. Simultaneous use of the fast interrupt and the DMAC II is not available.
5. Bits ILVL2 to ILVL0 in the interrupt control register should be set after the DMAII bit is set. The DMA II transfer is not affected by the I flag or the IPL.

Figure 10.5 Registers RIPL1 and RIPL2

10.6.4 Interrupt Sequence

The interrupt sequence is performed from when an interrupt request has been accepted until the interrupt handler starts.
For most instructions, when an interrupt request is generated while an instruction is being executed, the requested interrupt is evaluated in the priority resolver after the current instruction is completed. If appropriate, the interrupt sequence starts from the next cycle.
For instructions RMPA, SCMPU, SIN, SMOVB, SMOVF, SMOVU, SOUT, SSTR, SUNTIL, and SWHILE, as soon as an interrupt request is generated, the requested interrupt is evaluated suspending the current instruction being executed. If appropriate, the interrupt sequence starts immediately.
The interrupt sequence is as follows:

(1) The CPU acknowledges the interrupt request to obtain the interrupt information (the interrupt number, and the interrupt request level) from the interrupt controller. Then the corresponding IR bit becomes 0 (no interrupt requested).
(2) The state of the flag register (FLG) before the interrupt sequence is stored to a temporary register (1) in the CPU.
(3) The following bits in the flag register (FLG) become 0:
- The I flag (interrupt enable flag): interrupt disabled
- The D flag (debug flag): single-step interrupt disabled
- The U flag (stack pointer select flag): ISP selected

(4) The contents of the temporary register (1) in the CPU is saved to the stack; or to the save flag register (SVF) in case of the fast interrupt.
(5) The contents of the program counter (PC) is saved to the stack; or to the save PC register (SVP) in case of the fast interrupt.
(6) The interrupt request level for the accepted interrupt is set in the IPL (processor interrupt priority level).
(7) The corresponding interrupt vector is read from the interrupt vector table.
(8) This interrupt vector is stored into the program counter (PC).

When the interrupt sequence completes, the interrupt handler is initiated.

Note:
1. This register is inaccessible to users.

10.6.5 Interrupt Response Time

The interrupt response time, as shown in Figure 10.6, consists of two non-overlapping time segments: (a) the period from when an interrupt request is generated until the instruction being executed is completed; and (b) the period required for the interrupt sequence.

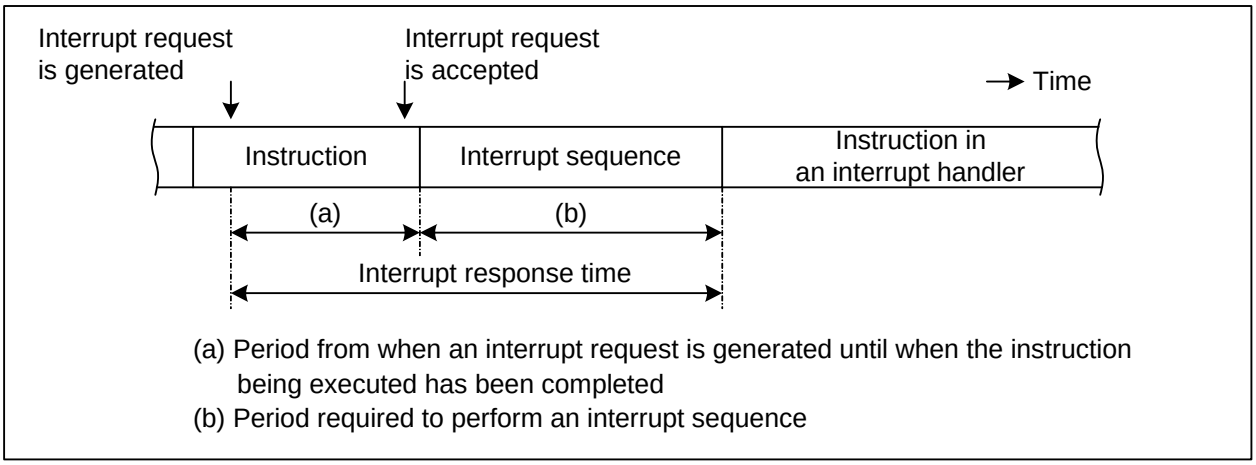

Interrupt request is generated
Interrupt request is accepted
Time
Instruction
Interrupt sequence
Instruction in an interrupt handler
(a)
(b)
Interrupt response time
(a) Period from when an interrupt request is generated until when the instruction being executed has been completed
(b) Period required to perform an interrupt sequence

Figure 10.6 Interrupt Response Time

Period (a) varies depending on the instruction being executed. Instructions, such as LDCTX and STCTX in which registers are sequentially saved/restored into/from the stack, require the longest time. For example, the STCTX instruction requires at least 30 cycles for ten registers to be saved. It requires more time if the WAIT instruction is in the stack.
Period (b) is listed in Table 10.7.

Table 10.7 Interrupt Sequence Execution Time (1)

Interrupt	Execution Time in Terms of CPU Clock
Peripheral	13 + α cycles (2)
INT instruction	11 cycles
NMI	10 cycles
Watchdog timer Oscillator stop detection	11 cycles
Undefined instruction	12 cycles
Overflow	12 cycles
BRK instruction (relocatable vector table)	16 cycles
BRK instruction (fixed vector table)	19 cycles
BRK2 instruction	19 cycles
Fast interrupt	11 cycles

Notes:
1. The interrupt vectors should be aligned in addresses in multiples of 4 of internal ROM. The fast interrupt is independent of this condition.
2. α is the number of waits to access SFR minus 2.

10.6.6 IPL After Interrupt Request Acceptance

When a peripheral interrupt request is accepted, the interrupt request level is set in the IPL (processor interrupt priority level).

Software interrupts and special interrupts have no interrupt request level. For these interrupt requests, if accepted, the value shown in Table 10.8 is set in the IPL as interrupt request level.

Table 10.8 Interrupts without Interrupt Request Level and IPL

Interrupt Sources without Interrupt Request Level	IPL Value to be Set
NMI, watchdog timer, oscillator stop detection	7
Reset	0
Software	Unchanged

10.6.7 Register Saving

In the interrupt sequence, the flag register (FLG) and program counter (PC) values are saved to the stack, in that order. Figure 10.7 shows the stack status before and after an interrupt request is accepted.

In the fast interrupt sequence, the flag register (FLG) and program counter (PC) values are saved to the save flag register (SVF) and save PC register (SVP), respectively.

If there are any other registers to be saved to the stack, save them at the beginning of the interrupt handler. A single PUSHM instruction saves all registers except the frame base register (FB) and stack pointer (SP).

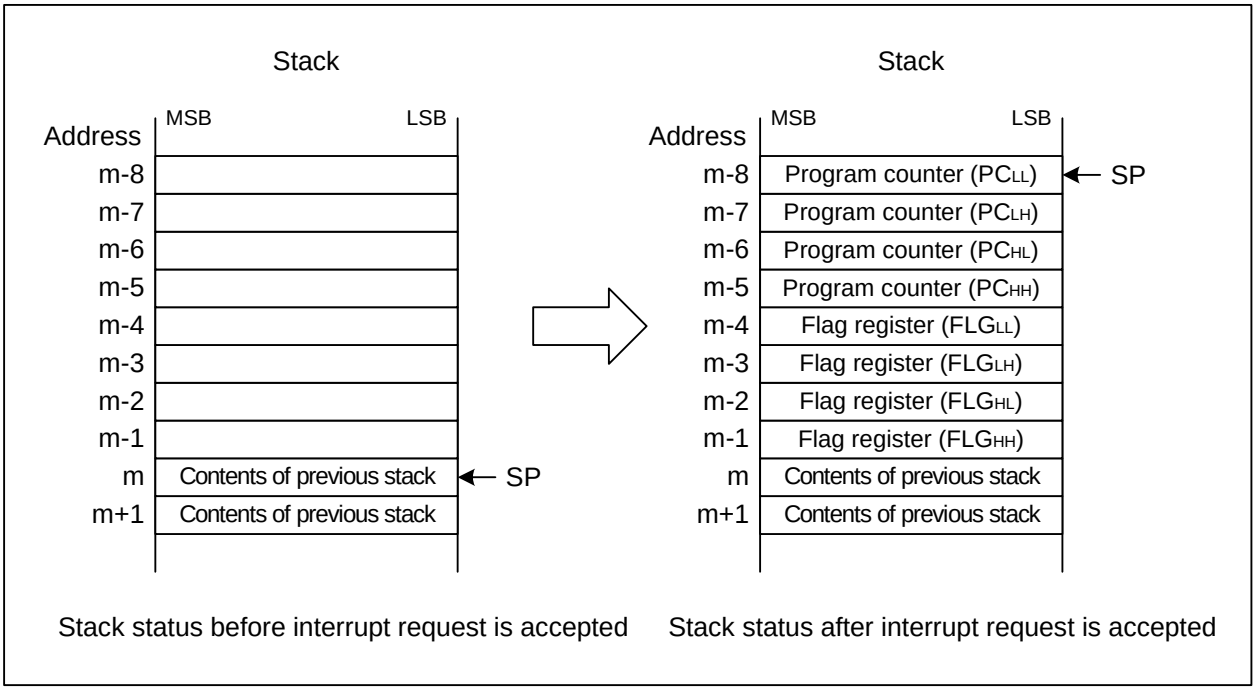


Figure 10.7 Stack Status Before and After an interrupt Request is Accepted

10.7 Register Restoring from Interrupt Handler

When the REIT instruction is executed at the end of the interrupt handler, the saved values of the flag register (FLG) and the program counter (PC) are restored from the stack, and the program resumes the operation that has been interrupted. In the fast interrupt, execute the FREIT instruction to restore them from the save registers, instead.

To restore the values of registers, which are saved by software in the interrupt handler, use an instruction such as POPM before the REIT or FREIT instruction.

If the register bank is switched in the interrupt handler, the bank is automatically switched back to the original register bank by the REIT or FREIT instruction.

10.8 Interrupt Priority

If two or more interrupt requests are detected at an interrupt request sampling point, the interrupt request with higher priority is accepted.

For maskable interrupts (peripheral interrupts), the interrupt request level select bits (bits ILVL2 to ILVL0) select a request level. If there are more than two interrupts with the same level, they are accepted according to their relative priority predetermined by the hardware.

The priorities of the reset and special interrupts, such as the watchdog timer interrupt, are determined by the hardware. Note that the reset has the highest priority. The following is the priority order of hardware interrupts:

$$\text{Reset} > \begin{matrix}\text{Watchdog timer}\\ \text{Oscillator stop detection}\end{matrix} > \text{NMI} > \text{Peripherals}$$

Software interrupts are not governed by priority. They always cause execution to jump to the interrupt handler whenever the relevant instruction is executed.

10.9 Priority Resolver

The priority resolver determines which interrupt request has a higher priority if two or more interrupt requests are detected at a sampling point.

Figure 10.8 shows the priority resolver.

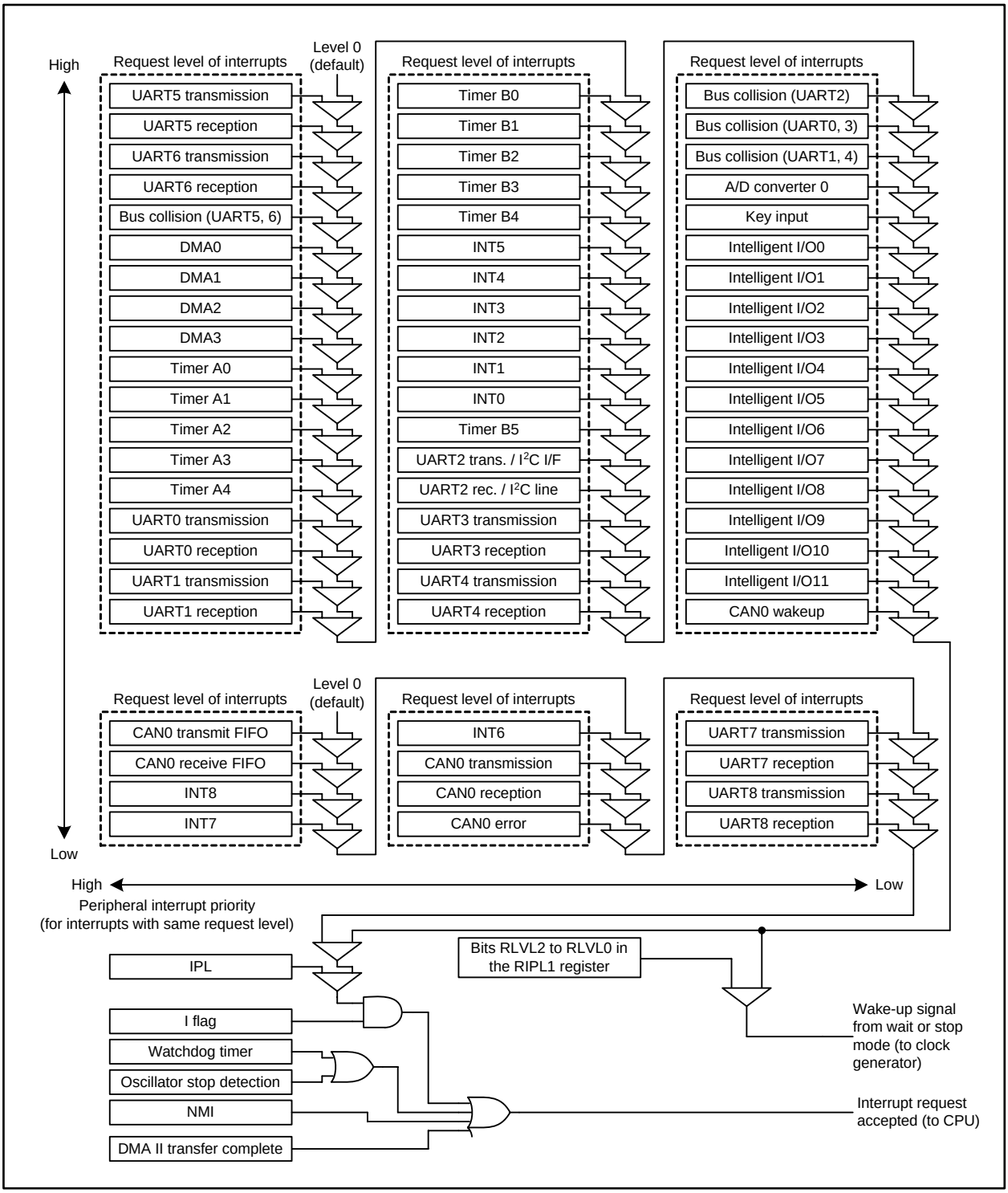


Figure 10.8 Priority Resolver

10.10 External Interrupt

An external interrupt is generated by an external input applied to the $\overline{INTi}$ pin (i = 0 to 8). The LVS bit in the INTiIC register selects whether an interrupt is triggered by the effective edge(s) (edge sensitive), or by the effective level (level sensitive) of the input signal. The polarity of the input signal is selected by the POL bit in the same register.

When using edge-triggered interrupts, setting the IFSR0j bit in the IFSR0 register to 1 (both edges) causes interrupt requests to be generated on both rising and falling edges of the external input applied to the $\overline{INTj}$ pin (j = 0 to 5). This also applies to setting the IFSR1n bit (n = m - 6) in the IFSR1 register to 1 (both edges) for the $\overline{INTm}$ pin (m = 6 to 8). When the IFSR0j bit or the IFSR1n bit is set to 1, the POL bit in the corresponding register should be set to 0 (falling edge).

When using level-triggered interrupts, set the IFSR0j or IFSR1n to 0 (one edge). When an effective level, which is selected by the POL bit, is detected on the $\overline{INTi}$ pin, the IR bit in the INTiIC register becomes 1. The IR bit remains unchanged until the INTi interrupt is accepted, or it is set to 0 by a program, even if the signal level at the $\overline{INTi}$ pin changes.

Figure 10.9 and Figure 10.10 show registers IFSR0 and IFSR1, respectively.

External Interrupt Request Source Select Register 0

b7 b6 b5 b4 b3 b2 b1 b0

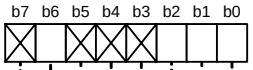
Symbol	Address	Reset Value
IFSR0	4406Fh	0000 0000b

Bit Symbol	Bit Name	Function	RW
IFSR00	$\overline{INT0}$ Pin Polarity Select Bit (1)	0: One edge 1: Both edges	RW
IFSR01	$\overline{INT1}$ Pin Polarity Select Bit (1)	0: One edge 1: Both edges	RW
IFSR02	$\overline{INT2}$ Pin Polarity Select Bit (1)	0: One edge 1: Both edges	RW
IFSR03	$\overline{INT3}$ Pin Polarity Select Bit (1)	0: One edge 1: Both edges	RW
IFSR04	$\overline{INT4}$ Pin Polarity Select Bit (1)	0: One edge 1: Both edges	RW
IFSR05	$\overline{INT5}$ Pin Polarity Select Bit (1)	0: One edge 1: Both edges	RW
IFSR06	UART0/UART3 Interrupt Source Select Bit	0: Bus collision, start condition detection, stop condition detection in UART3 1: Bus collision, start condition detection, stop condition detection in UART0	RW
IFSR07	UART1/UART4 Interrupt Source Select Bit	0: Bus collision, start condition detection, stop condition detection in UART4 1: Bus collision, start condition detection, stop condition detection in UART1	RW

Note:
1. This bit should be set to 0 to select the level sensitive input as trigger. To set this bit to 1, the POL bit in the corresponding INTiIC register (i = 0 to 5) should be set to 0 (falling edge).

Figure 10.9 IFSR0 Register

External Interrupt Request Source Select Register 1



Symbol	Address	Reset Value
IFSR1	4406Dh	X0XX X000b

Bit Symbol	Bit Name	Function	RW
IFSR10	$\overline{\text{INT6}}$ Pin Polarity Select Bit (1)	0: One edge 1: Both edges	RW
IFSR11	$\overline{\text{INT7}}$ Pin Polarity Select Bit (1)	0: One edge 1: Both edges	RW
IFSR12	$\overline{\text{INT8}}$ Pin Polarity Select Bit (1)	0: One edge 1: Both edges	RW
— (b5-b3)	No register bits; should be written with 0 and read as undefined value		—
IFSR16	UART5/UART6 Interrupt Source Select Bit	0: Bus collision, start condition detection, stop condition detection in UART5 1: Bus collision, start condition detection, stop condition detection in UART6	RW
— (b7)	No register bit; should be written with 0 and read as undefined value		—

Note:
1. This bit should be set to 0 to select the level sensitive input as trigger. To set this bit to 1, the POL bit in the corresponding INTiIC register (i = 6 to 8) should be set to 0 (falling edge).

Figure 10.10 IFSR1 Register

10.11 NMI

The NMI (Non Maskable Interrupt) occurs when an input signal at the $\overline{\text{NMI}}$ pin switches from high to low. This non maskable interrupt is disabled after a reset. To enable this interrupt, the PM24 bit in the PM2 register should be set to 1 after setting the interrupt stack pointer (ISP) at the beginning of the program. The $\overline{\text{NMI}}$ pin shares a pin with the port P8_5, which enables the P8_5 bit in the P8 register to indicate the input level at the $\overline{\text{NMI}}$ pin.

Note:
1. When not using the NMI, hold 0 as reset value of the PM24 bit in the PM2 register.

10.12 Key Input Interrupt

The key input interrupt is enabled by setting ports P10_4 to P10_7 as input ports.

The interrupt request is generated if any of the signals applied to ports P10_4 to P10_7 switches from high to low. This interrupt also functions as key wake-up to exit wait or stop mode. Figure 10.11 shows a block diagram of the key input interrupt. If any of the ports is held low, signals applied to other ports are not detected as interrupt request signals.

To use the key input interrupt, every register from P10_4S to P10_7S should be set to 00h (I/O port) and bits PD10_4 to PD10_7 should be set to 0 (input). This is the only setting available for the key input interrupt.

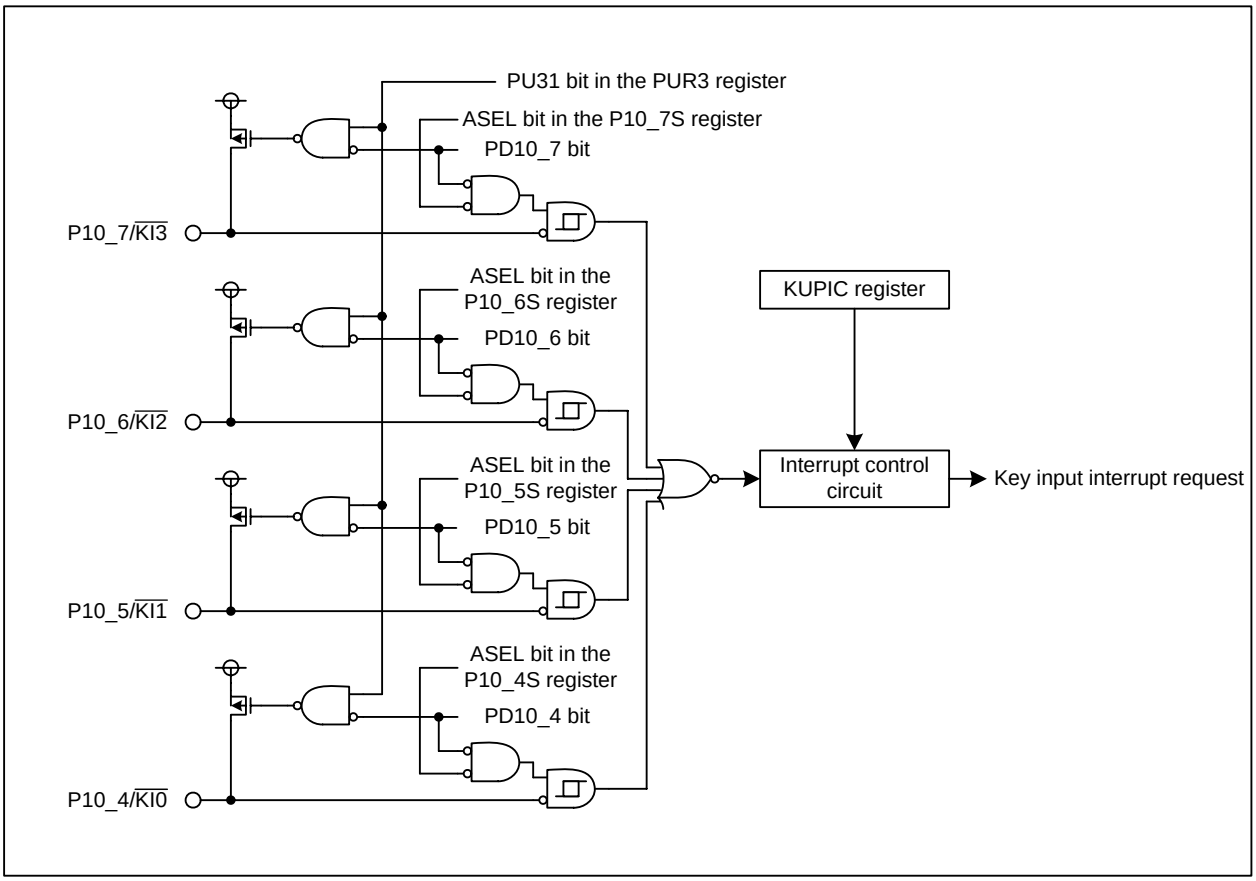


Figure 10.11 Key Input Interrupt

10.13 Intelligent I/O Interrupt

The intelligent I/O interrupt is assigned to software interrupt numbers from 44 to 55.

Figure 10.12 shows a block diagram of the intelligent I/O interrupt. Figure 10.13 and Figure 10.14 show registers IIOiIR and IIOiIE (i = 0 to 11), respectively.

To use the intelligent I/O interrupt, the IRLT bit in the IIOiIE register should be set to 1 (interrupt requests used for interrupt).

The intelligent I/O interrupt contains various request sources. When an interrupt request is generated with an intelligent I/O function, the corresponding bit in the IIOiIR register becomes 1 (interrupts requested). If the corresponding bit in the IIOiIE register is set to 1 (interrupt enabled), the IR bit in the corresponding IIOiIC register changes to 1 (interrupts requested).

After the IR bit setting changes from 0 to 1, this bit remains unchanged if a bit in the IIOiIR register is set to 1 by another interrupt request source and the corresponding bit in the IIOiIE register is set to 1.

Bits in the IIOiIR register are not set to 0 automatically even if an interrupt is accepted. They should be set to 0 by either the AND or BCLR instruction. Note that every generated interrupt request is ignored until these bit are set to 0.

To use the intelligent I/O interrupt to activate DMAC II, the IRLT bit in the IIOiIE register should be set to 0 (interrupt requests used for DMA or DMA II) and the bit for interrupt source to be used in the IIOiE register should be set to 1 (interrupt enabled).

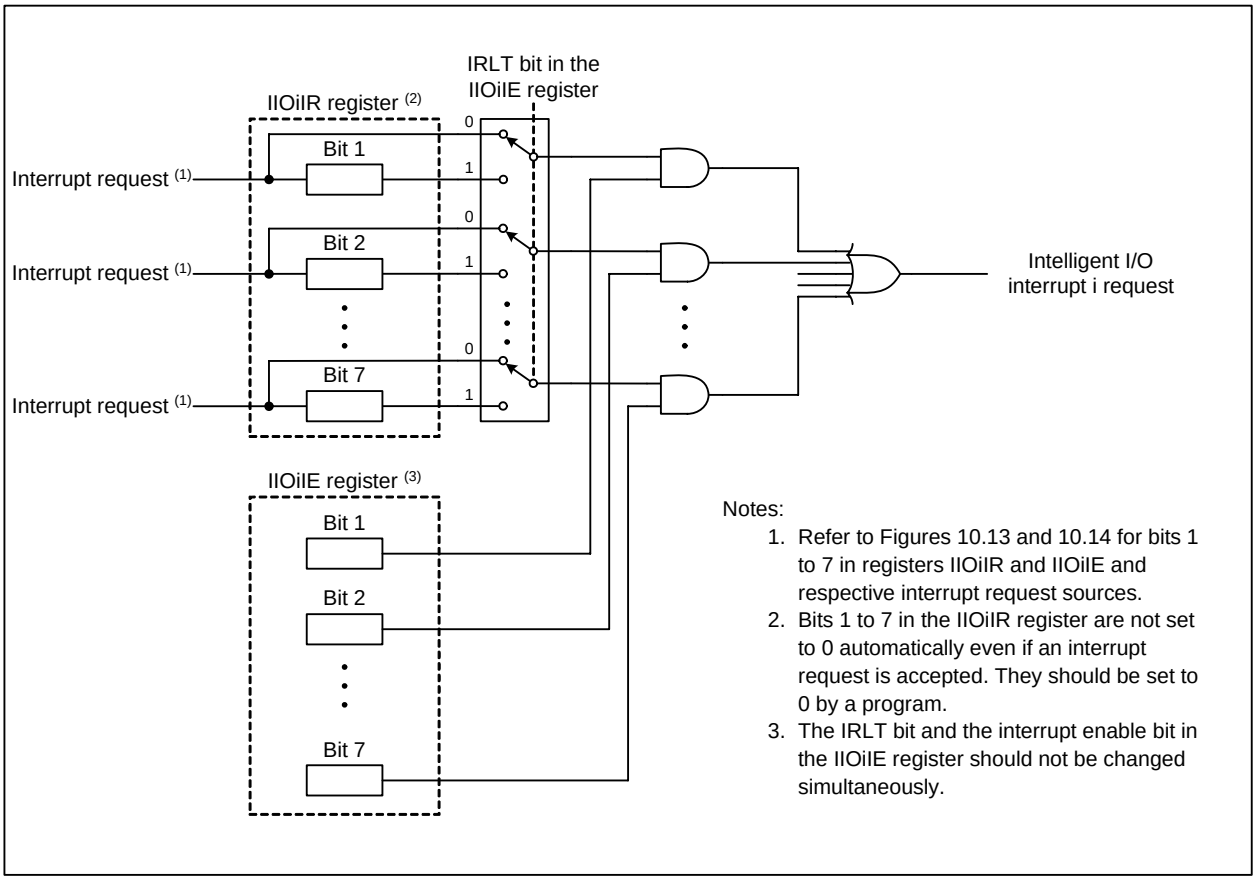
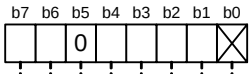

IRLT bit in the IIOiIE register
IIOiIR register (2)
Bit 1
Bit 2
Bit 7
Interrupt request (1)
Interrupt request (1)
Interrupt request (1)
0
1
0
1
0
1
Intelligent I/O interrupt i request
IIOiIE register (3)
Bit 1
Bit 2
Bit 7
Notes:
1. Refer to Figures 10.13 and 10.14 for bits 1 to 7 in registers IIOiIR and IIOiIE and respective interrupt request sources.
2. Bits 1 to 7 in the IIOiIR register are not set to 0 automatically even if an interrupt request is accepted. They should be set to 0 by a program.
3. The IRLT bit and the interrupt enable bit in the IIOiIE register should not be changed simultaneously.

Figure 10.12 Intelligent I/O Interrupt Block Diagram (i = 0 to 11)

Intelligent I/O Interrupt Request Register i (i = 0 to 11)



Symbol: IIO0IR to IIO11IR
Address: Refer to the table below
Reset Value: ??0? ???1b (1)

Bit Symbol	Bit Name	Function	RW
— (b0)	No register bit; this bit is read as 1		—
(Note 2)	0: No interrupt requested 1: Interrupt requested (3)		RW
(Note 2)	0: No interrupt requested 1: Interrupt requested (3)		RW
(Note 2)	0: No interrupt requested 1: Interrupt requested (3)		RW
(Note 2)	0: No interrupt requested 1: Interrupt requested (3)		RW
— (b5)	Reserved	Should be written with 0	RW
(Note 2)	0: No interrupt requested 1: Interrupt requested (3)		RW
(Note 2)	0: No interrupt requested 1: Interrupt requested (3)		RW

Notes:
1. When the register has any function-assigned bit, the reset value is X (undefined); otherwise, the reset value is 0.
2. Refer to the table below for bit symbols.
3. When this bit is function-assigned, it can be set to 0 only. It should not be set to 1. To set to 0, either the AND or BCLR instruction should be used; when the bit is not function-assigned, that is, reserved, it should be set to 0.

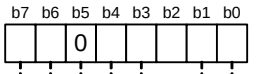
Bit Symbols for the Intelligent I/O Interrupt Request Register

Symbol	Address	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
IIO0IR	00A0h	—	—	—	—	—	TM13R/PO13R	TM02R/PO02R	—
IIO1IR	00A1h	—	—	—	—	—	TM14R/PO14R	TM00R/PO00R	—
IIO2IR	00A2h	—	—	—	—	—	TM12R/PO12R	—	—
IIO3IR	00A3h	—	—	—	—	PO27R	TM10R/PO10R	TM03R/PO03R	—
IIO4IR	00A4h	—	—	—	BT1R	—	TM17R/PO17R	TM04R/PO04R	—
IIO5IR	00A5h	—	—	—	SIO2RR	—	PO21R	TM05R/PO05R	—
IIO6IR	00A6h	—	—	—	SIO2TR	—	PO20R	TM06R/PO06R	—
IIO7IR	00A7h	IE0R	—	—	BT0R	—	PO22R	TM07R/PO07R	—
IIO8IR	00A8h	IE1R	IE2R	—	BT2R	—	PO23R	TM11R/PO11R	—
IIO9IR	00A9h	—	INT6R	—	—	—	PO24R	TM15R/PO15R	—
IIO10IR	00AAh	—	INT7R	—	—	—	PO25R	TM16R/PO16R	—
IIO11IR	00ABh	—	INT8R	—	—	—	PO26R	TM01R/PO01R	—

BTxR : Intelligent I/O group x base timer interrupt request (x = 0 to 2)
TMxyR : Intelligent I/O group x time measurement channel y interrupt request (x = 0, 1; y = 0 to 7)
POxyR : Intelligent I/O group x waveform generation channel y interrupt request (x = 0 to 2; y = 0 to 7)
IEzR : Intelligent I/O group 2 IEBus interrupt request (z = 0 to 2)
SIO2RR : Intelligent I/O group 2 receive interrupt request
SIO2TR : Intelligent I/O group 2 transmit interrupt request
INTmR : INTm interrupt request (m = 6 to 8)

Figure 10.13 Registers IIO0IR to IIO11IR

Intelligent I/O Interrupt Enable Register i (i = 0 to 11)



Symbol	Address	Reset Value
IIO0IE to IIO11IE	Refer to the table below.	0000 0000b

Bit Symbol	Bit Name	Function	RW
IRLT	Interrupt Request Select Bit (2)	0: Use interrupt requests for DMA or DMA II 1: Use interrupt requests for interrupt	RW
(Note 1)	0: Disable the interrupt of bit 1 in the IIOiIR register 1: Enable the interrupt of bit 1 in the IIOiIR register		RW
(Note 1)	0: Disable the interrupt of bit 2 in the IIOiIR register 1: Enable the interrupt of bit 2 in the IIOiIR register		RW
(Note 1)	0: Disable the interrupt of bit 3 in the IIOiIR register 1: Enable the interrupt of bit 3 in the IIOiIR register		RW
(Note 1)	0: Disable the interrupt of bit 4 in the IIOiIR register 1: Enable the interrupt of bit 4 in the IIOiIR register		RW
— (b5)	Reserved	Should be written with 0	RW
(Note 1)	0: Disable the interrupt of bit 6 in the IIOiIR register 1: Enable the interrupt of bit 6 in the IIOiIR register		RW
(Note 1)	0: Disable the interrupt of bit 7 in the IIOiIR register 1: Enable the interrupt of bit 7 in the IIOiIR register		RW

Notes:
1. Refer to the table below for bit symbols.
2. To use interrupt requests for interrupt, the IRLT bit should be set to 1, then bits 1 to 4, 6, and 7 should be set to 1.

Bit Symbols for the Intelligent I/O Interrupt Enable Register

Symbol	Address	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
IIO0IE	00B0h	—	—	—	—	—	TM13E/PO13E	TM02E/PO02E	IRLT
IIO1IE	00B1h	—	—	—	—	—	TM14E/PO14E	TM00E/PO00E	IRLT
IIO2IE	00B2h	—	—	—	—	—	TM12E/PO12E	—	IRLT
IIO3IE	00B3h	—	—	—	—	PO27E	TM10E/PO10E	TM03E/PO03E	IRLT
IIO4IE	00B4h	—	—	—	BT1E	—	TM17E/PO17E	TM04E/PO04E	IRLT
IIO5IE	00B5h	—	—	—	SIO2RE	—	PO21E	TM05E/PO05E	IRLT
IIO6IE	00B6h	—	—	—	SIO2TE	—	PO20E	TM06E/PO06E	IRLT
IIO7IE	00B7h	IE0E	—	—	BT0E	—	PO22E	TM07E/PO07E	IRLT
IIO8IE	00B8h	IE1E	IE2E	—	BT2E	—	PO23E	TM11E/PO11E	IRLT
IIO9IE	00B9h	—	INT6E	—	—	—	PO24E	TM15E/PO15E	IRLT
IIO10IE	00BAh	—	INT7E	—	—	—	PO25E	TM16E/PO16E	IRLT
IIO11IE	00BBh	—	INT8E	—	—	—	PO26E	TM01E/PO01E	IRLT

BTxE : Intelligent I/O group x base timer interrupt enabled (x = 0 to 2)
TMxyE : Intelligent I/O group x time measurement channel y interrupt enabled (x = 0, 1; y = 0 to 7)
POxyE : Intelligent I/O group x waveform generation channel y interrupt enabled (x = 0 to 2; y = 0 to 7)
IEzE : Intelligent I/O group 2 IEBus interrupt enabled (z = 0 to 2)
SIO2RE : Intelligent I/O group 2 receive interrupt enabled
SIO2TE : Intelligent I/O group 2 transmit interrupt enabled
INTmE : INTm interrupt enabled (m = 6 to 8)

Figure 10.14 Registers IIO0IE to IIO11IE

10.14 Notes on Interrupts

10.14.1 ISP Setting

The interrupt stack pointer (ISP) is initialized to 00000000h after a reset. Set a value to the ISP before an interrupt is accepted, otherwise the program may go out of control. A multiple of 4 should be set to the ISP, which enables faster interrupt sequence due to less memory access.
For the use of NMI, in particular, since this interrupt cannot be disabled, the PM24 bit in the PM2 register should be set to 1 (NMI enabled) after the ISP is set at the beginning of program.

10.14.2 NMI

- The NMI cannot be disabled once the PM24 bit in the PM2 register is set to 1 (NMI enabled). This bit setting should be done only for the use of NMI.

- When the PM24 bit in the PM2 register is set to 1 (NMI enabled), the P8_5 bit in the P8 register is enabled just for monitoring the $\overline{\text{NMI}}$ pin state. It is not enabled as a general port.

10.14.3 External Interrupt

- The input signal to the $\overline{\text{INTi}}$ pin (i = 0 to 8) requires the pulse width specified by the electrical characteristics. If a pulse width is narrower than the specification, the external interrupt may not be accepted.

- When the effective level and/or edge of $\overline{\text{INTi}}$ pin (i = 0 to 8) are/is changed by the following bits: bits POL and/or LVS in the INTiIC register, the IFSR0i bit (i = 0 to 5) in the IFSR0 register, and/or the IFSR1j bit (j = i - 6; i = 6 to 8) in the IFSR1 register, the corresponding IR bit may become 1 (interrupt requested). When setting the above mentioned bits, preset bits ILVL2 to ILVL0 in the INTiIC register to 000b (interrupt disabled). After setting the above mentioned bits, set the corresponding IR bit to 0 (no interrupt requested), then set bits ILVL2 to ILVL0.

- The interrupt input signals to pins $\overline{\text{INT6}}$ to $\overline{\text{INT8}}$ are also connected to bits INT6R to INT8R in registers IIO9IR to IIO11IR. Therefore, these input signals, when assigned to the intelligent I/O, can be used as a source for exiting wait mode or stop mode. Note that these signals are enabled only on the falling edge and not affected by the following bit settings: bits POL and LVS in the INTiIC register (i = 0 to 8), IFSR0i bit (i = 0 to 5) in the IFSR0 register, and the IFSR1j bit (j = i - 6; i = 6 to 8) in the IFSR1 register.

11. Watchdog Timer

The watchdog timer monitors program executions and detects defective programs. The 15-bit watchdog counter counts downward with the cycle which is the peripheral bus clock frequency divided by the prescaler.
When the watchdog timer underflows, the CM06 bit in CM0 register selects either a watchdog timer interrupt request or a reset. Once the CM06 bit is set to 1 (reset), it cannot be changed to 0 (watchdog timer interrupt) by a program. Only after a reset, it can be set to 0.
The watchdog timer contains a prescaler which is the peripheral bus clock divided by 16 or 128. The divide ratio is selected by setting the WDC7 bit in the WDC register.
The watchdog timer is stopped in wait mode, stop mode, or when the $\overline{\text{HOLD}}$ is driven low. It resumes counting from the value held when the mode or state is exited.

The general formula to calculate a watchdog timer period is:

$$\text{Watchdog timer period} = \frac{\text{Prescaler divider factor (16 or 128)} \times 32768}{\text{Peripheral bus clock frequency}}$$

For example, when the peripheral bus clock is 1/2 of 50 MHz-CPU clock and the prescaler has a divide-by-16 operation, the watchdog timer period is approximately 21 ms. Note that marginal errors within one prescaler output cycle may occur in the watchdog timer period.

The watchdog timer is initialized when a write to the WDTS register is performed or when a watchdog timer interrupt request is generated. The prescaler is initialized only when the MCU is reset.
After a reset, both the watchdog timer and the prescaler are stopped. They start counting when a write to the WDTS register is performed.

Figure 11.1 shows a block diagram of the watchdog timer. Figure 11.2 and Figure 11.3 show registers associated with the watchdog timer.

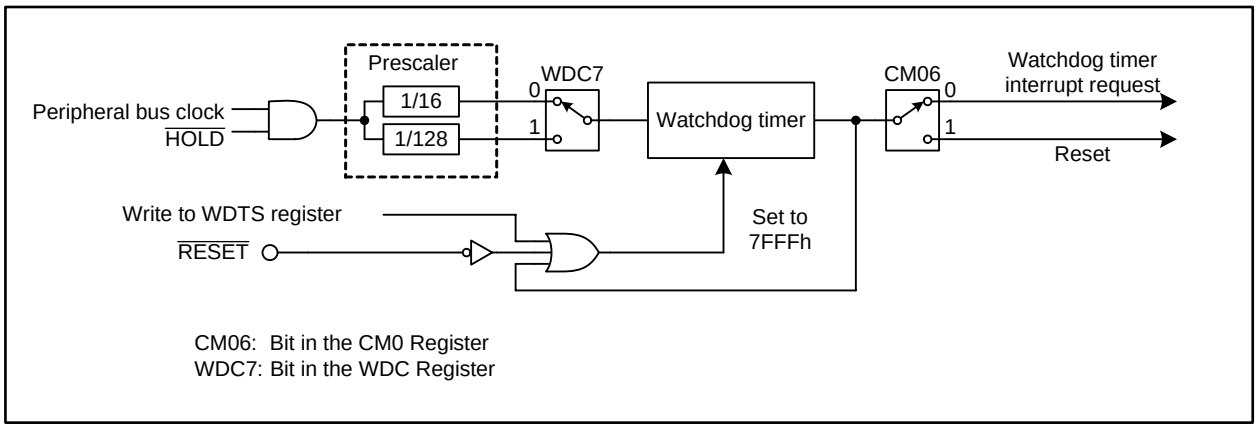


Figure 11.1 Watchdog Timer Block Diagram

Watchdog Timer Control Register

b7	b6	b5	b4	b3	b2	b1	b0
	0	0					

Symbol	Address	Reset Value
WDC	4404Fh	000X XXXXb

Bit Symbol	Bit Name	Function	RW
— (b4-b0)	Upper five bits of the watchdog timer (b14 to b10)		RO
— (b6-b5)	Reserved	Should be written with 0	RW
WDC7	Prescaler Select Bit (1)	0: Divide-by-16 1: Divide-by-128	RW

Note:
1. Set this bit before activating the watchdog timer.

Figure 11.2 WDC Register

Watchdog Timer Start Register

b7	b0

Symbol	Address	Reset Value
WDTS	4404Eh	Undefined

Function	RW
The watchdog timer is initialized by a write access. Then it starts counting downward. 7FFFh is set as the default value by writing a given value	WO

Figure 11.3 WDTS Register

12. DMAC

Direct Memory Access (DMA) is a system that can control data transfer without using the CPU.
The R32C/100 Series' four channel DMA controller (DMAC) transmits 8-bit (byte), 16-bit (word), or 32-bit (long word) data in cycle-steal mode from a source address to a destination address every time a transfer request is generated.
The DMAC, which shares a data bus with the CPU, has a higher bus access priority than the CPU. This allows the DMAC to perform fast data transfer when a transfer request is generated.
Figure 12.1 shows a map of the CPU-internal registers associated with DMAC. Table 12.1 lists DMAC specifications. Figure 12.2 to Figure 12.10 show registers associated with DMAC. Since the registers shown in Figure 12.1 are allocated in the CPU, the LDC or STC instruction should be used to write to the registers.

DMAC-associated Registers

DMD0	DMA0 mode register
DMD1	DMA1 mode register
DMD2	DMA2 mode register
DMD3	DMA3 mode register
DCT0	DMA0 terminal count register
DCT1	DMA1 terminal count register
DCT2	DMA2 terminal count register
DCT3	DMA3 terminal count register
DCR0	DMA0 terminal count reload register (1)
DCR1	DMA1 terminal count reload register (1)
DCR2	DMA2 terminal count reload register (1)
DCR3	DMA3 terminal count reload register (1)
DSA0	DMA0 source address register
DSA1	DMA1 source address register
DSA2	DMA2 source address register
DSA3	DMA3 source address register
DSR0	DMA0 source address reload register (1)
DSR1	DMA1 source address reload register (1)
DSR2	DMA2 source address reload register (1)
DSR3	DMA3 source address reload register (1)
DDA0	DMA0 destination address register
DDA1	DMA1 destination address register
DDA2	DMA2 destination address register
DDA3	DMA3 destination address register
DDR0	DMA0 destination address reload register (1)
DDR1	DMA1 destination address reload register (1)
DDR2	DMA2 destination address reload register (1)
DDR3	DMA3 destination address reload register (1)

Note:
1. Registers are used for repeat transfer, not for single transfer.

Figure 12.1 CPU-internal Registers for DMAC

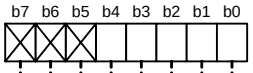
Table 12.1 DMAC Specifications

Item		Specification
Channels		4
Bus request mode		Cycle-steal mode
Transfer memory spaces		From a given address in a 64-Mbyte space (00000000h to 01FFFFFFh and FE000000h to FFFFFFFFh) to another given address in the same space
Maximum transfer bytes		64-Mbytes (when 32-bit data is transferred), 32-Mbytes (when 16-bit data is transferred), 16-Mbytes (when 8-bit data is transferred)
DMA request sources (1)		Falling edge or both edges of signals applied to pins INT0 to INT3 or pins $\overline{\text{INT6}}$ to $\overline{\text{INT8}}$ Timers A0 to A4 interrupt requests Timers B0 to B5 interrupt requests UART0 to UART8 transmit/receive interrupt requests A/D conversion interrupt requests Intelligent I/O interrupt requests Multi-master I^2C-bus interrupt requests Software trigger
Channel priority		DMA0 > DMA1 > DMA2 > DMA3 (DMA0 has the highest priority)
Transfer sizes		8 bits, 16 bits, or 32 bits
Addressing modes		Incrementing addressing or non-incrementing addressing
Transfer modes	Single transfer	Transfer is completed when the DCTi register (i = 0 to 3) is set to 00000000h
	Repeat transfer	When the DCTi register is set to 00000000h, the value of the DCRi register is reloaded into the DCTi register to continue the DMA transfer
DMA transfer complete interrupt request generation timing		When the DCTi register changes from 00000001h to 00000000h
DMA transfer start-up	Single transfer	DMA starts when a DMA transfer request is generated after the DCTi register is set to a value more than 00000001h and bits MDi1 and MDi0 in the DMDi register are set to 01b (single transfer)
	Repeat transfer	DMA starts when a DMA transfer request is generated after the DCTi register is set to a value more than 00000001h and bits MDi1 and MDi0 are set to 11b (repeat transfer)
DMA transfer stop	Single transfer	DMA stops when bits MDi1 and MDi0 are set to 00b (DMA disabled)
	Repeat transfer	DMA stops when bits MDi1 and MDi0 are set to 00b (DMA disabled)
Reload timing to DCTi, DSAi or DDAi register		When the DCTi register changes from 00000001h to 00000000h in repeat transfer mode
Minimum DMA transfer cycles		3

Note:
1. DMA transfer does not affect each interrupt.

The DMA transfer request is available by two different sources: software and hardware. More concretely, they are a write access to the DSR bit in the DMiSL2 register (i = 0 to 3) and an interrupt request output from a function specified in bits DSEL4 to DSEL0 in the DMiSL register, and in bits DSEL24 to DSEL20 in the DMiSL2 register. Unlike interrupt requests, the DMA transfer request is not affected by the I flag nor the interrupt control register. Therefore this request can be accepted even when any interrupt request cannot be because of “interrupt disabled”. Since the DMA transfer does not affect any interrupt, either, the IR bit in the interrupt control register is not changed by the DMA transfer.

DMAi Request Source Select Register (i = 0 to 3)



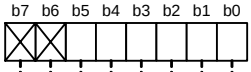
Symbol	Address	Reset Value
DM0SL to DM3SL	44078h, 44079h, 4407Ah, 4407Bh	XXX0 0000b

Bit Symbol	Bit Name	Function	RW
DSEL0	DMA Request Source Select Bit (1)	Refer to Table 12.2 “DMiSL Register (i = 0 to 3) Functions”	RW
DSEL1			RW
DSEL2			RW
DSEL3			RW
DSEL4			RW
— (b7-b5)	No register bits; should be written with 0 and read as undefined value		—

Note:
1. The bit settings of bits DSEL4 to DSEL0 should be changed while bits MDi1 and MDi0 in the DMDi register of the corresponding channel are set to 00b (DMA transfer disabled).

Figure 12.2 Registers DM0SL to DM3SL

DMAi Request Source Select Register 2 (i = 0 to 3)



Symbol	Address	Reset Value
DM0SL2 to DM3SL2	44070h, 44071h, 44072h, 44073h	XX00 0000b

Bit Symbol	Bit Name	Function	RW
DSEL20	DMA Request Source Select Bit (1)	Refer to Table 12.3 "DMiSL2 Register (i = 0 to 3) Functions"	RW
DSEL21			RW
DSEL22			RW
DSEL23			RW
DSEL24			RW
DSR	Software DMA Transfer Request Bit	When a software trigger is selected, a DMA transfer request is generated by setting this bit to 1 (The bit is read as 0)	RW
— (b7-b6)	No register bits; should be written with 0 and read as undefined value		—

Note:
1. The bit settings of bits DSEL24 to DSEL20 should be changed while bits MDi1 and MDi0 in the DMDi register of the corresponding channel are set to 00b (DMA transfer disabled).

Figure 12.3 Registers DM0SL2 to DM3SL2

Table 12.2 DMiSL Register (i = 0 to 3) Functions

Setting Value	DMA Request Source			
b4 b3 b2 b1 b0	DMA0	DMA1	DMA2	DMA3
0 0 0 0 0	Select from DMiSL2 register			
0 0 0 0 1	Falling edge of $\overline{INT0}$ (1)	Falling edge of $\overline{INT1}$ (1)	Falling edge of $\overline{INT2}$ (1)	Falling edge of $\overline{INT3}$ (1)
0 0 0 1 0	Both edges of $\overline{INT0}$ (1)	Both edges of $\overline{INT1}$ (1)	Both edges of $\overline{INT2}$ (1)	Both edges of $\overline{INT3}$ (1)
0 0 0 1 1	Timer A0 interrupt request			
0 0 1 0 0	Timer A1 interrupt request			
0 0 1 0 1	Timer A2 interrupt request			
0 0 1 1 0	Timer A3 interrupt request			
0 0 1 1 1	Timer A4 interrupt request			
0 1 0 0 0	Timer B0 interrupt request			
0 1 0 0 1	Timer B1 interrupt request			
0 1 0 1 0	Timer B2 interrupt request			
0 1 0 1 1	Timer B3 interrupt request			
0 1 1 0 0	Timer B4 interrupt request			
0 1 1 0 1	Timer B5 interrupt request			
0 1 1 1 0	UART0 transmit interrupt request			
0 1 1 1 1	UART0 receive interrupt request or ACK interrupt request (2)			
1 0 0 0 0	UART1 transmit interrupt request			
1 0 0 0 1	UART1 receive interrupt request or ACK interrupt request (2)			
1 0 0 1 0	UART2 transmit interrupt request or I^2C-bus interface interrupt request (3)			
1 0 0 1 1	UART2 receive interrupt request, ACK interrupt request (2), or I^2C-bus line interrupt request (3)			
1 0 1 0 0	UART3 transmit interrupt request		UART5 transmit interrupt request	
1 0 1 0 1	UART3 receive interrupt request or ACK interrupt request (2)		UART5 receive interrupt request or ACK interrupt request (2)	
1 0 1 1 0	UART4 transmit interrupt request		UART6 transmit interrupt request	
1 0 1 1 1	UART4 receive interrupt request or ACK interrupt request (2)		UART6 receive interrupt request or ACK interrupt request (2)	
1 1 0 0 0	A/D0 interrupt request			
1 1 0 0 1	Intelligent I/O interrupt 0 request	Intelligent I/O interrupt 7 request	Intelligent I/O interrupt 2 request	Intelligent I/O interrupt 9 request
1 1 0 1 0	Intelligent I/O interrupt 1 request	Intelligent I/O interrupt 8 request	Intelligent I/O interrupt 3 request	Intelligent I/O interrupt 10 request
1 1 0 1 1	Intelligent I/O interrupt 2 request	Intelligent I/O interrupt 9 request	Intelligent I/O interrupt 4 request	Intelligent I/O interrupt 11 request
1 1 1 0 0	Intelligent I/O interrupt 3 request	Intelligent I/O interrupt 10 request	Intelligent I/O interrupt 5 request	Intelligent I/O interrupt 0 request
1 1 1 0 1	Intelligent I/O interrupt 4 request	Intelligent I/O interrupt 11 request	Intelligent I/O interrupt 6 request	Intelligent I/O interrupt 1 request
1 1 1 1 0	Intelligent I/O interrupt 5 request	Intelligent I/O interrupt 0 request	Intelligent I/O interrupt 7 request	Intelligent I/O interrupt 2 request
1 1 1 1 1	Intelligent I/O interrupt 6 request	Intelligent I/O interrupt 1 request	Intelligent I/O interrupt 8 request	Intelligent I/O interrupt 3 request

Notes:

1. The falling edge and both edges of signals applied to the $\overline{INTi}$ pin (i = 0 to 3) cause a DMA request generation. The external interrupts (bits POL and LVS in the INTiIC register and the IFSR0 register) are not affected by these DMA request sources, and vice versa.
2. Registers UiSMR and UiSMR2 (i = 0 to 6) are used to switch between the UARTi receive interrupt and ACK interrupt.
3. Select an interrupt source either of UART2 or I^2C-bus interface by using the I2CEN bit in the I2CMR register.

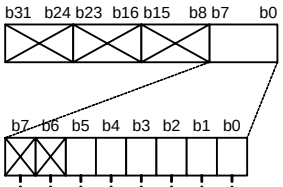
Table 12.3 DMiSL2 Register (i = 0 to 3) Functions

Setting Value	DMA Request Source			
b4 b3 b2 b1 b0	DMA0	DMA1	DMA2	DMA3
0 0 0 0 0	Software trigger			
0 0 0 0 1	Falling edge of $\overline{\text{INT6}}$ (1)	Falling edge of $\overline{\text{INT7}}$ (1)	Falling edge of $\overline{\text{INT8}}$ (1)	Reserved
0 0 0 1 0	Both edges of $\overline{\text{INT6}}$ (1)	Both edges of $\overline{\text{INT7}}$ (1)	Both edges of $\overline{\text{INT8}}$ (1)	Reserved
0 0 0 1 1	Reserved			
0 0 1 0 0	Reserved			
0 0 1 0 1	Reserved			
0 0 1 1 0	Reserved			
0 0 1 1 1	Reserved			
0 1 0 0 0	Reserved			
0 1 0 0 1	Reserved			
0 1 0 1 0	Reserved			
0 1 0 1 1	Reserved			
0 1 1 0 0	Reserved			
0 1 1 0 1	Reserved			
0 1 1 1 0	Reserved			
0 1 1 1 1	Reserved			
1 0 0 0 0	Reserved			
1 0 0 0 1	Reserved			
1 0 0 1 0	Reserved			
1 0 0 1 1	Reserved			
1 0 1 0 0	Reserved			
1 0 1 0 1	Reserved			
1 0 1 1 0	Reserved			
1 0 1 1 1	Reserved			
1 1 0 0 0	UART7 transmit interrupt request			
1 1 0 0 1	UART7 receive interrupt request			
1 1 0 1 0	UART8 transmit interrupt request			
1 1 0 1 1	UART8 receive interrupt request			
1 1 1 0 0	Reserved			
1 1 1 0 1	Reserved			
1 1 1 1 0	Reserved			
1 1 1 1 1	Reserved			

Note:

1. The falling edge and both edges of signals applied to the $\overline{\text{INTi}}$ pin (i = 6 to 8) cause a DMA request generation. The external interrupts (bits POL and LVS in the INTiIC register and the IFSR1 register) are not affected by these DMA request sources, and vice versa.

DMAi Mode Register (i = 0 to 3) (1)



Symbol: DMD0 to DMD3

Address: (CPU internal register)

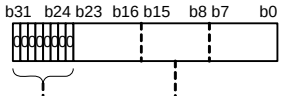
Reset Value: XXXX XXXX XXXX XXXX XXXX XXXX XX00 0000b

Bit Symbol	Bit Name	Function	RW
MDi0	Transfer Mode Select Bit (2)	b1 b0 0 0 : DMA transfer disabled 0 1 : Single transfer 1 0 : Do not use this combination 1 1 : Repeat transfer	RW
MDi1			RW
BWi0	Transfer Size Select Bit (3)	b3 b2 0 0 : 8 bits 0 1 : 16 bits 1 0 : 32 bits 1 1 : Do not use this combination	RW
BWi1			RW
USAi	Source Addressing Mode Select Bit (3)	0: Non-incrementing addressing 1: Incrementing addressing	RW
UDAi	Destination Addressing Mode Select Bit (3)	0: Non-incrementing addressing 1: Incrementing addressing	RW
— (b7-b6)	No register bits; should be written with 0 and read as undefined value		—
— (b31-b8)	No register bits; should be written with 0 and read as undefined value		—

Notes:
1. The LDC instruction should be used to write to this register.
2. This bit should be set after all other DMAC-associated registers are set.
3. Set bits MDi1 and MDi0 to 00b before rewriting these bits.

Figure 12.4 Registers DMD0 to DMD3

DMAi Terminal Count Register (i = 0 to 3) (1)



Symbol: DCT0 to DCT3

Address: (CPU internal register)

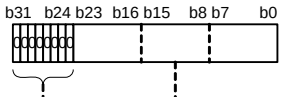
Reset Value: XXXX XXXXh

Function	Setting Range	RW
Set the transfers to be performed	000000h to FFFFFFh (2)	RW
Reserved	Should be set to 00h	RW

Notes:
1. The LDC instruction should be used to write to this register. The register should be set while bits MDi1 and MDi0 in the DMDi register of the corresponding channel are set to 00b (DMA transfer disabled).
2. When this register is set to 000000h, any new DMA transfer request is unacceptable.

Figure 12.5 Registers DCT0 to DCT3

DMAi Terminal Count Reload Register (i = 0 to 3) (1)



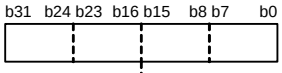
Symbol	Address	Reset Value
DCR0 to DCR3	(CPU internal register)	XXXX XXXXh

Function	Setting Range	RW
Set the transfers to be performed	000000h to FFFFFFh	RW
Reserved	Should be set to 00h	RW

Note:
1. The LDC instruction should be used to write to this register. The register should be set while bits MDi1 and MDi0 in the DMDi register of the corresponding channel are set to 00b (DMA transfer disabled).

Figure 12.6 Registers DCR0 to DCR3

DMAi Source Address Register (i = 0 to 3) (1)



Symbol	Address	Reset Value
DSA0 to DSA3	(CPU internal register)	XXXX XXXXh

Function	Setting Range	RW
Set a source address	00000000h to 01FFFFFFh and FE000000h to FFFFFFFFh (64-Mbyte space)	RW

Note:
1. The LDC instruction should be used to write to this register. The register should be set while bits MDi1 and MDi0 in the DMDi register of the corresponding channel are set to 00b (DMA transfer disabled).

Figure 12.7 Registers DSA0 to DSA3

DMAi Source Address Reload Register (i = 0 to 3) (1)

b31 b24	b23 b16	b15 b8	b7 b0

Symbol	Address	Reset Value
DSR0 to DSR3	(CPU internal register)	XXXX XXXXh

Function	Setting Range	RW
Set a source address	00000000h to 01FFFFFFh and FE000000h to FFFFFFFFh (64-Mbyte space)	RW

Note:
1. The LDC instruction should be used to write to this register. The register should be set while bits MDi1 and MDi0 in the DMDi register of the corresponding channel are set to 00b (DMA transfer disabled).

Figure 12.8 Registers DSR0 to DSR3

DMAi Destination Address Register (i = 0 to 3) (1)

b31 b24	b23 b16	b15 b8	b7 b0

Symbol	Address	Reset Value
DDA0 to DDA3	(CPU internal register)	XXXX XXXXh

Function	Setting Range	RW
Set a destination address	00000000h to 01FFFFFFh and FE000000h to FFFFFFFFh (64-Mbyte space)	RW

Note:
1. The LDC instruction should be used to write to this register. The register should be set while bits MDi1 and MDi0 in the DMDi register of the corresponding channel are set to 00b (DMA transfer disabled).

Figure 12.9 Registers DDA0 to DDA3

DMAi Destination Address Reload Register (i = 0 to 3) (1)

b31 b24	b23 b16	b15 b8	b7 b0

Symbol	Address	Reset Value
DDR0 to DDR3	(CPU internal register)	XXXX XXXXh

Function	Setting Range	RW
Set a destination address	00000000h to 01FFFFFFh and FE000000h to FFFFFFFFh (64-Mbyte space)	RW

Note:
1. The LDC instruction should be used to write to this register. The register should be set while bits MDi1 and MDi0 in the DMDi register of the corresponding channel are set to 00b (DMA transfer disabled).

Figure 12.10 Registers DDR0 to DDR3

12.1 Transfer Cycle

The transfer cycle is composed of bus cycles to read data from memory or SFR (source read) and to write data to destination address (destination write).
The read and write bus cycles vary with the setting of registers DSAi (i = 0 to 3) and DDAi, the width of data bus connected to the relevant device and bus timing.

12.1.1 Effect of Transfer Address and Data Bus Width

Table 12.4 lists the incremental bus cycles caused by transfer address alignment or data bus width.

Table 12.4 Incremental Bus Cycles Caused by Transfer Address and Data Bus Width

Transfer Data Unit	Data Bus Width	Transfer Address	Bus Cycles to be Incremented	Bus Cycles Generated
8-bit transfer	8 to 64 bits	n	0	[n]
16-bit transfer	8 bits	n	+1	[n] - [n + 1]
	16 bits	2n	0	[2n]
		2n + 1	+1	[2n + 1] - [2n + 2]
	32 bits	4n	0	[4n]
		4n + 1	0	[4n + 1]
		4n + 2	0	[4n + 2]
		4n + 3	+1	[4n + 3] - [4n + 4]
	64 bits	8n	0	[8n]
		8n + 1	0	[8n + 1]
		8n + 2	0	[8n + 2]
		8n + 3	0	[8n + 3]
		8n + 4	0	[8n + 4]
		8n + 5	0	[8n + 5]
		8n + 6	0	[8n + 6]
		8n + 7	+1	[8n + 7] - [8n + 8]
32-bit transfer	8 bits	n	+3	[n] - [n + 1] - [n + 2] - [n + 3]
	16 bits	4n	+1	[4n] - [4n + 2]
		4n + 1	+2	[4n + 1] - [4n + 2] - [4n + 4]
		4n + 2	+1	[4n + 2] - [4n + 4]
		4n + 3	+2	[4n + 3] - [4n + 4] - [4n + 6]
	32 bits	4n	0	[4n]
		4n + 1	+1	[4n + 1] - [4n + 4]
		4n + 2	+1	[4n + 2] - [4n + 4]
		4n + 3	+1	[4n + 3] - [4n + 4]
	64 bits	8n	0	[8n]
		8n + 1	0	[8n + 1]
		8n + 2	0	[8n + 2]
		8n + 3	0	[8n + 3]
		8n + 4	0	[8n + 4]
		8n + 5	+1	[8n + 5] - [8n + 8]
		8n + 6	+1	[8n + 6] - [8n + 8]
		8n + 7	+1	[8n + 7] - [8n + 8]

12.1.2 Effect of Bus Timing

In the R32C/100 Series, each device has its own bus addresses assigned. The bus width and bus timing vary with each device. Table 12.5 lists the bus width and access cycles for each device.

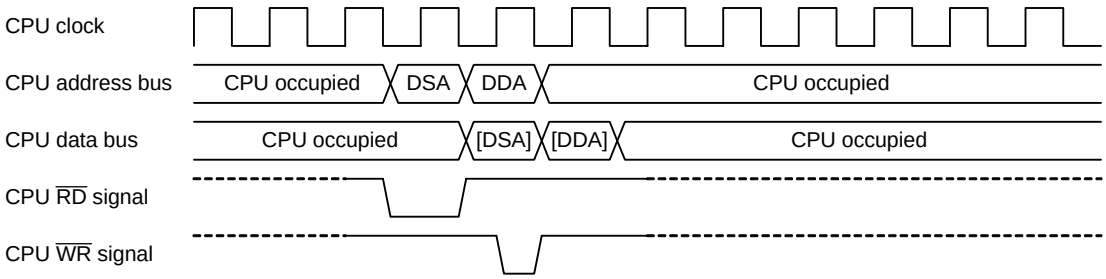
Table 12.5 Bus Width and Bus Cycles

Device	Addresses (1)	Bus Width	Access Cycles (2)	Reference Clock
Flash memory	FFE00000h to FFFFFFFFh	64-bit	2 or 3 (3)	CPU clock
Data flash	00060000h to 00061FFFh	64-bit	5	CPU clock
RAM	00000400h to 0003FFFFh	64-bit	1 or 2 (4)	CPU clock
SFR space	00000000h to 0000001Fh	16-bit	3 (5)	Peripheral bus clock
	00000020h to 000003FFh	16-bit	2 (5)	Peripheral bus clock
SFR2 space	00040000h to 00041FFFh	16-bit	2 (5)	Peripheral bus clock
	00042000h to 00043FFFh	32-bit	2 (5)	Peripheral bus clock
	00044000h to 000440DFh	16-bit	2 (5, 6)	Peripheral bus clock
	000440E0h to 000443FFh	16-bit	3 (5, 6)	Peripheral bus clock
	00044400h to 00045FFFh	16-bit	2 (5, 6)	Peripheral bus clock
	00046000h to 000467FFh	32-bit	3 (5, 6)	Peripheral bus clock
	00046800h to 00047FFFh	32-bit	2 (5, 6)	Peripheral bus clock
	00048000h to 0004FFFFh	64-bit	2	CPU clock

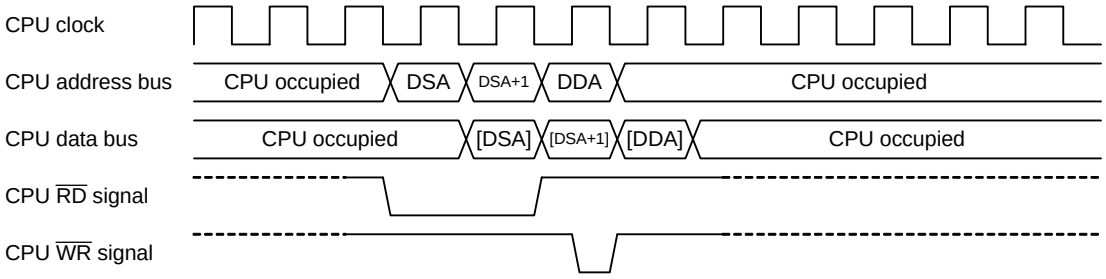
Notes:
1. Reserved spaces are included.
2. Access cycles are based on each bus clock.
3. An access to the same page as the previous time requires two cycles. Otherwise, three cycles are required.
4. If write cycles are generated sequentially, each write cycle except the initial one has two access cycles. A read cycle just after a write cycle has also two access cycles.
5. If SFR is sequentially accessed, each access except the initial one has additional one base clock cycle.
6. One or less access cycle may be added depending on the phase of peripheral bus clock.

Figure 12.11 shows an example of source-read bus cycles in a transfer cycle. In this figure, the number of source-read bus cycle is shown under different conditions, provided that the destination address is in an internal RAM with one bus cycle of destination-write. In real operation, the transfer cycles change according to conditions for destination-write bus cycles as well as for source-read bus cycles. To calculate a transfer cycle, therefore, respective conditions should be applied to both destination-write bus cycle and source-read bus cycle. In (2) of Figure 12.11, for example, if the destination-write bus cycle is generated twice, both bus cycles are two, respectively.

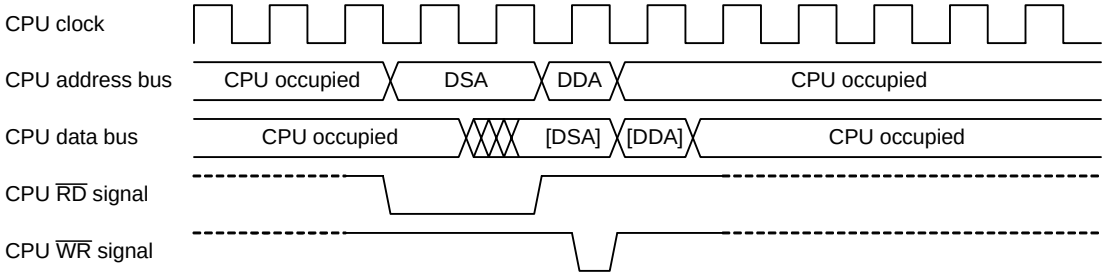
(1) One bus cycle of source-read is generated
Example: 16-bit data transfer from the address 8n of the RAM



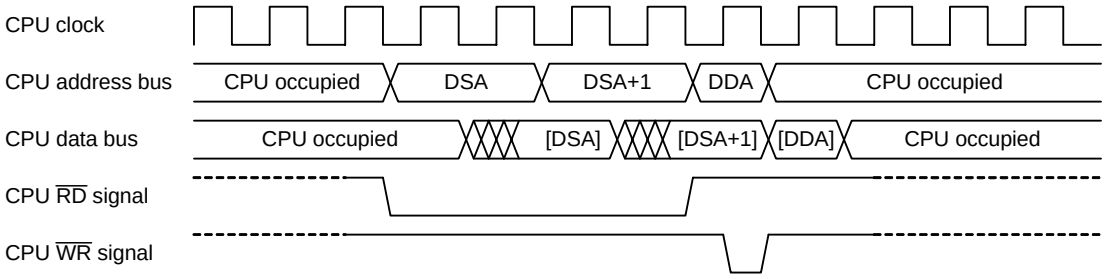
(2) Two bus cycles of source-read are generated
Example: 16-bit data transfer from the address 8n+7 of the RAM



(3) One bus cycle of source-read is generated with one wait cycle
Example: 16-bit data transfer from the address 16n of the ROM



(4) Two bus cycles of source-read is generated with one wait cycle
Example: 16-bit data transfer from the address 16n+7 of the ROM



Note:
1. The above applies under the condition of one bus cycle of destination-write. In real cases, the number of destination-write bus cycles should be considered according to the conditions such as above.

Figure 12.11 Source-read Bus Cycles in a Transfer Cycle

12.2 DMA Transfer Cycle

The DMA transfer cycles are calculated as follows:

$$\textit{Number of a transfer cycles} = \textit{Source-read bus cycles} \times j + \textit{Destination-write bus cycles} \times k + 1$$

where:

j = access cycles for read,

k = access cycles for write (refer to Table 12.5)

Each bus cycle, source-read, and destination-write basically requires one or more cycles. In addition, more cycles may be required depending on the transfer address. Refer to Table 12.4 for required bus cycles.

"+1" in the formula above means a cycle required to decrement the value of DCTi register (i = 0 to 3).

The following are calculation examples:

To transfer 32-bit data from the address 400h of the RAM to the address 800h of the RAM,

$$\textit{Number of the transfer cycles} = 1 \times 1 + 1 \times 1 + 1$$
$$= 3$$

Thus, there are three cycles.

To transfer 16-bit data from the AD00 register at address 380h to registers P1 and P0 at addresses 3C1h and 3C0h, respectively, with the peripheral bus clock (= 1/2 CPU clock),

$$\textit{Number of the transfer cycles} = 1 \times 2 \times 2 + 1 \times 2 \times 2 + 1$$
$$= 9$$

Thus, there are nine cycles.

12.3 Channel Priority and DMA Transfer Timing

When multiple DMA transfer requests are generated in the same sampling period, between the falling edge of the CPU clock and the next falling edge, these requests are simultaneously input into the DMAC. Channel priority in this case is: DMA0 > DMA1 > DMA2 > DMA3.

Figure 12.12 shows an example of the DMA transfer by external source, specifically when a DMA0 request and a DMA1 request are simultaneously generated. The DMA0 request having higher priority is received first to start a transfer. After one DMA0 transfer is completed, the bus privilege is returned to the CPU. When the CPU has completed one bus access, the DMA1 transfer starts. After one DMA1 transfer is completed, the privilege is again returned to the CPU.

DMA transfer requests cannot be counted up. The transfer occurs only once even when an $\overline{\text{INTi}}$ interrupt is generated more than once before receiving the bus privilege, as the DMA1 shown in Figure 12.12.

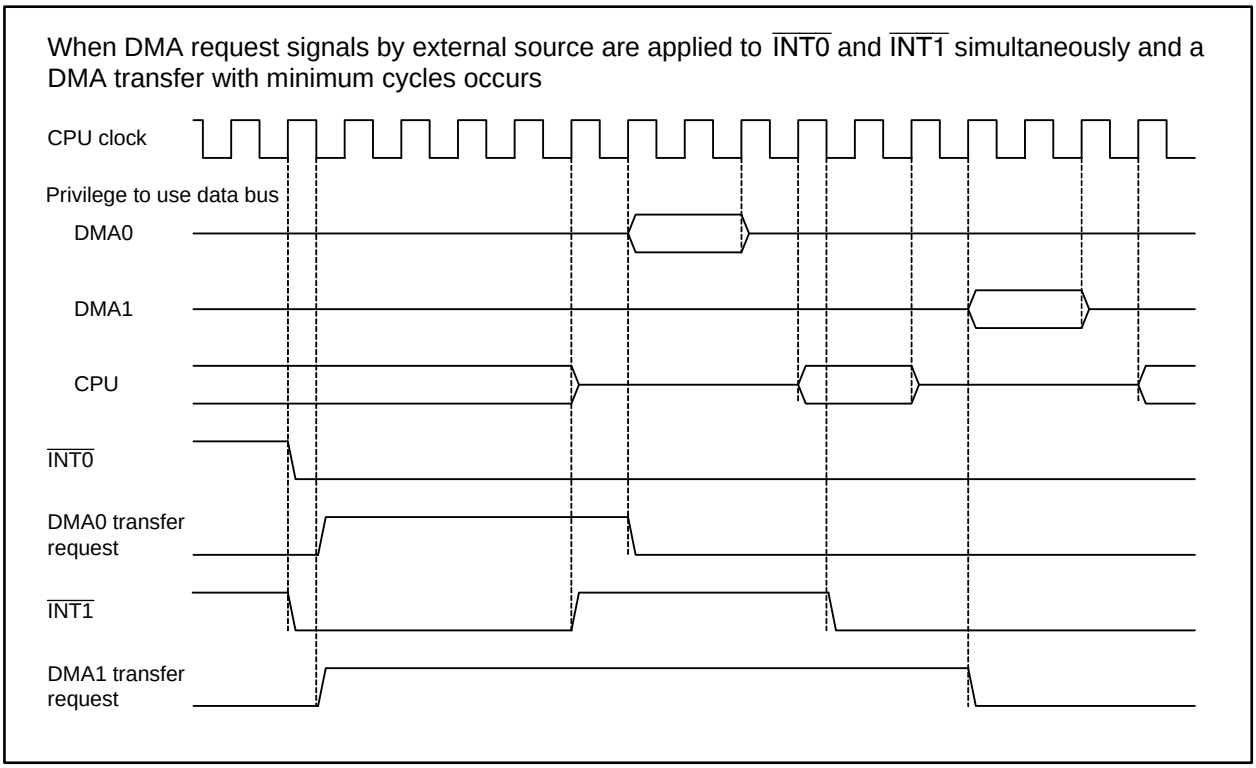

When DMA request signals by external source are applied to INT0 and INT1 simultaneously and a DMA transfer with minimum cycles occurs
CPU clock
Privilege to use data bus
DMA0
DMA1
CPU
INT0
DMA0 transfer request
INT1
DMA1 transfer request

Figure 12.12 DMA Transfer by External Source

12.4 Notes on DMAC

12.4.1 DMAC-associated Register Settings

- Set the DMAC-associated registers while bits MDi1 and MDi0 (i = 0 to 3) in the DMDi register are 00b (DMA transfer disabled). Then, set bits MDi1 and MDi0 to 01b (single transfer) or 11b (repeat transfer) at the end of the setup procedure. This procedure is also applied to rewriting bits UDAi, USAi, and BWi1 and BWi0 in the DMDi register.

- In case the DMAC-associated registers are to be rewritten while DMA transfer is enabled, disable the peripheral function as DMA request source so that no DMA transfer request is generated, then set bits MDi1 and MDi0 in the DMDi register of the corresponding channel to 00b (DMA transfer disabled).

- Once a DMA transfer request is accepted, DMA transfer cannot be disabled even if setting bits MDi1 and MDi0 in the DMDi register to 00b (DMA transfer disabled). Do not change the settings of any DMAC-associated registers other than bits MDi1 and MDi0 until the DMA transfer is completed.

- Wait six or more peripheral bus clocks to set bits MDi1 and MDi0 in the DMDi register to 01b (single transfer) or 11b (repeat transfer) after setting registers DMiSL and DMiSL2.

12.4.2 Read from DMAC-associated Registers

- To sequentially read respective registers DMiSL and DMiSL2, follow the reading order as below:
  DM0SL, DM1SL, DM2SL, and DM3SL
  DM0SL2, DM1SL2, DM2SL2, and DM3SL2

13. DMAC II

DMAC II is activated by an interrupt request from any peripheral function, and performs data transfer without a CPU instruction. Transfer sources can be selected from memory, immediate data, memory + memory, and immediate data + memory.
Table 13.1 lists specifications of DMAC II.

Table 13.1 DMAC II Specifications

Item	Specification
Triggers for DMAC II	Interrupt requests generated by any of peripheral functions when bits ILVL2 to ILVL0 in the corresponding interrupt control register are set to 111b (level 7)
Transfer types	• Data in memory is transferred to memory (memory-to-memory transfer) • Immediate data is transferred to memory (immediate data transfer) • Data in memory + data in memory are transferred to memory (calculation transfer) • Immediate data + data in memory are transferred to memory (calculation transfer)
Transfer sizes	8 bits or 16 bits
Transfer memory spaces	From a given address in a 64-Mbyte space (00000000h to 01FFFFFFh and FE000000h to FFFFFFFFh) to another given address in the same space (1)
Addressing modes	Individually selectable for each source address and destination address from the following two modes: • Non-incrementing addressing: Address is held constant throughout a data transfer/a DMA II transaction • Incrementing addressing: Address increments by 1 (when 8-bit data is transferred) or 2 (when 16-bit data is transferred) after each data transfer
Transfer modes	• Single transfer: Only one data transfer is performed by one transfer request • Burst transfer: Data transfers are continuously performed for the number of times set in the transfer counter • Multiple transfer: Multiple memory-to-memory transfers are performed from different source addresses to different destination addresses by one transfer request
Chained transfer	Data transfer is sequentially performed according to a DMAC II Index (transfer information) linked with the previous transfer
DMA II transfer complete interrupt request	An interrupt request is generated when the transfer counter reaches 0000h

Note:
1. When 16-bit data is transferred to destination address at FFFFFFFFh, it is transferred to 00000000h as well as FFFFFFFFh. The same transfer is performed when the source address is FFFFFFFh.

13.1 DMAC II Settings

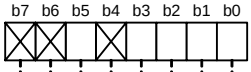
To activate DMAC II, set up the following items:
- Registers RIPL1 and RIPL2
- DMAC II index
- The interrupt control register of the peripheral function triggering DMAC II
- The relocatable vector of the peripheral function triggering DMAC II
- IIRLT bit in the IIOiIE register (i = 0 to 11) if the intelligent I/O interrupt is used. Refer to 10. “Interrupts” for details on the IIOilE register.

13.1.1 Registers RIPL1 and RIPL2

When the DMAII bits in both the RIPL1 and RIPL2 registers are set to 1 (DMA II transfer selected) and the FSIT bits are set to 0 (normal interrupt selected), DMAC II is activated by an interrupt of any peripheral function with bits ILVL2 to ILVL0 in the corresponding interrupt control register set to 111b (level 7).

Figure 13.1 shows registers RIPL1 and RIPL2.

Wake-up IPL Setting Register i (i = 1, 2) (1)



Symbol	Address	Reset Value
RIPL1, RIPL2	4407Fh, 4407Dh	XX0X 0000b

Bit Symbol	Bit Name	Function	RW
RLVL0	Interrupt Priority Level for Wake-up Select Bit (2)	b2 b1 b0 0 0 0 : Level 0 0 0 1 : Level 1 0 1 0 : Level 2 0 1 1 : Level 3 1 0 0 : Level 4 1 0 1 : Level 5 1 1 0 : Level 6 1 1 1 : Level 7	RW
RLVL1			RW
RLVL2			RW
FSIT	Fast Interrupt Select Bit (3)	0: Use interrupt request level 7 for normal interrupt 1: Use interrupt request level 7 for fast interrupt (4)	RW
— (b4)	No register bit; should be written with 0 and read as undefined value		—
DMAII	DMA II Select Bit (5)	0: Use interrupt request level 7 for interrupt 1: Use interrupt request level 7 for DMA II transfer (4)	RW
— (b7-b6)	No register bits; should be written with 0 and read as undefined value		—

Notes:
1. Registers RIPL1 and RIPL2 should be identically set.
2. The MCU exits wait mode or stop mode if the request level of requested interrupt is higher than the level selected using bits RLVL2 to RLVL0. These bits should be set to the same value as the IPL in the flag register (FLG).
3. When the FSIT bit is set to 1, an interrupt with interrupt request level 7 becomes the fast interrupt. In this case only one interrupt should be set to the interrupt request level 7.
4. Either the FSIT or DMAII bit should be set to 1. Simultaneous use of the fast interrupt and the DMAC II is not available.
5. Bits ILVL2 to ILVL0 in the interrupt control register should be set after the DMAII bit is set. The DMA II transfer is not affected by the I flag or the IPL.

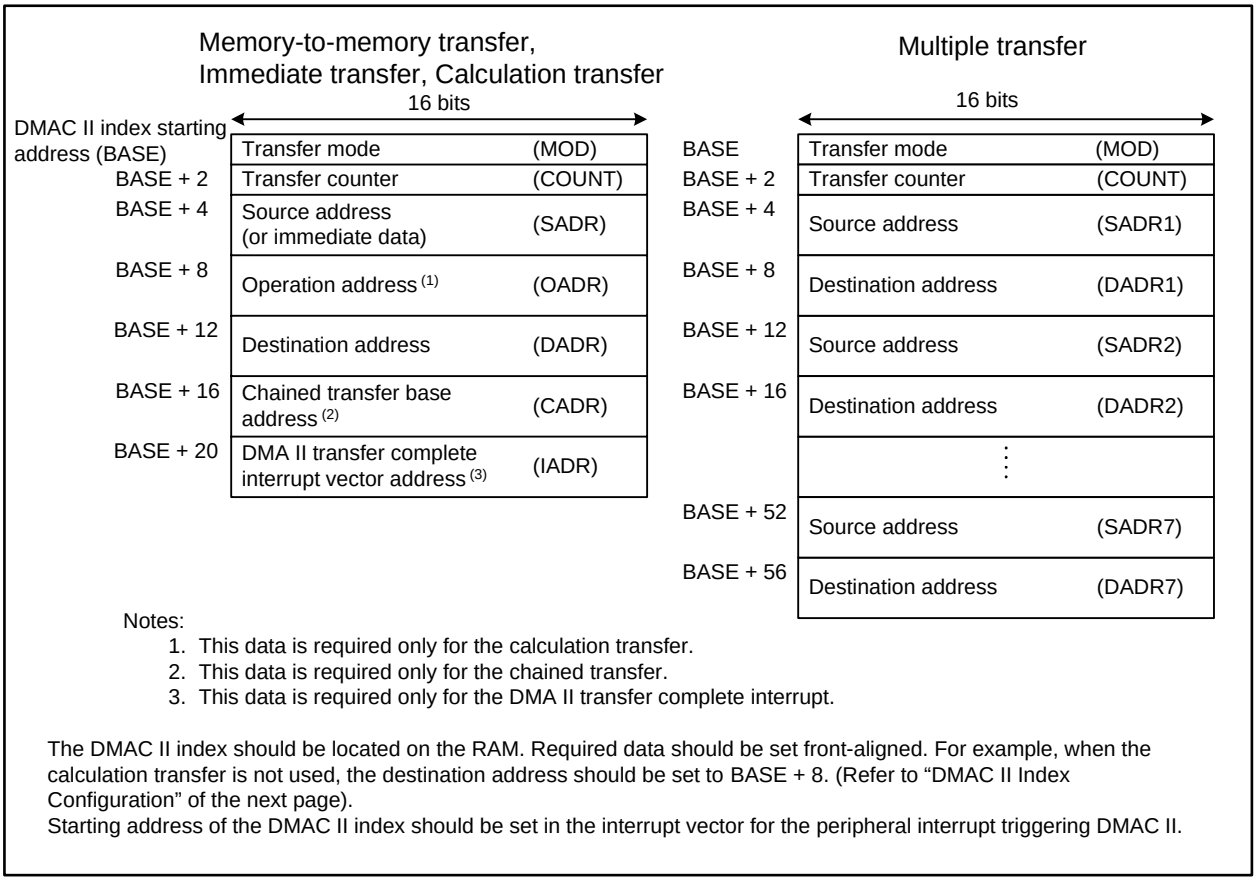
Figure 13.1 Registers RIPL1 and RIPL2

13.1.2 DMAC II Index

The DMAC II index is a data table of 12 to 60 bytes. It stores parameters for transfer mode, transfer counter, source address (or immediate data), operation address as an address to be calculated, destination address, chained transfer base address, and DMA II transfer complete interrupt vector address.

This DMAC II index should be located on the RAM.

Figure 13.2 shows a configuration of the DMAC II index and Table 13.2 lists a configuration example of the DMAC II index.



Notes:
1. This data is required only for the calculation transfer.
2. This data is required only for the chained transfer.
3. This data is required only for the DMA II transfer complete interrupt.

The DMAC II index should be located on the RAM. Required data should be set front-aligned. For example, when the calculation transfer is not used, the destination address should be set to BASE + 8. (Refer to “DMAC II Index Configuration” of the next page).
Starting address of the DMAC II index should be set in the interrupt vector for the peripheral interrupt triggering DMAC II.

Figure 13.2 DMAC II Index

The following are the details on the DMAC II index. These parameters should be aligned in the specified order listed in Table 13.2 according to the transfer mode to be performed.

- Transfer mode (MOD)
  2-byte data is required to set transfer mode. Figure 13.3 shows a configuration for transfer mode.
- Transfer counter (COUNT)
  2-byte data is required to set the transfers to be performed.
- Source address (SADR)
  4-byte data is required to set a source address in a memory or an immediate data. However, the two upper bytes of immediate data are ignored.
- Operation address (OADR)
  4-byte data is required to set an address in a memory to be calculated. This data setting is required only for the calculation transfer.
- Destination address (DADR)
  4-byte data is required to set a destination address in a memory.
- Chained transfer base address (CADR)
  4-byte data is required to set BASE, the starting address of the DMAC II index for the next transfer. This data setting is required only for the chained transfer.
- DMA II transfer complete interrupt vector address (IADR)
  4-byte data is required to set a jump address for the DMA II transfer complete interrupt handler. This data setting is required only for the DMA II transfer complete interrupt.

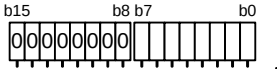
The symbols above are hereinafter used in place of their respective parameters.

Table 13.2 DMAC II Index Configuration

Transfer Data	Memory-to-memory Transfer/ Immediate Data Transfer				Calculation Transfer				Multiple Transfer
Chained transfer	Not used	Used	Not used	Used	Not used	Used	Not used	Used	Not available
DMA II transfer complete interrupt	Not used	Not used	Used	Used	Not used	Not used	Used	Used	Not available
DMAC II index	MOD COUNT SADR DADR 12 bytes	MOD COUNT SADR DADR CADR 16 bytes	MOD COUNT SADR DADR IADR 16 bytes	MOD COUNT SADR DADR CADR IADR 20 bytes	MOD COUNT SADR OADR DADR 16 bytes	MOD COUNT SADR OADR DADR CADR 20 bytes	MOD COUNT SADR OADR DADR IADR 20 bytes	MOD COUNT SADR OADR DADR CADR IADR 24 bytes	MOD COUNT SADR1 DADR1 SADRi DADRi i = 1 to 7 max. 60 bytes (when i = 7)

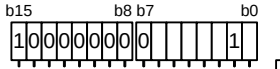
Transfer Mode (MOD) (1)

When multiple transfer is not selected (MULT = 0)



Bit Symbol	Bit Name	Function	RW
SIZE	Transfer Size Select Bit	0: 8 bits 1: 16 bits	RW
IMM	Transfer Source Select Bit	0: Immediate data 1: Memory	RW
UPDS	Source Addressing Select Bit	0: Non-incrementing addressing 1: Incrementing addressing	RW
UPDD	Destination Addressing Select Bit	0: Non-incrementing addressing 1: Incrementing addressing	RW
OPER	Calculation Transfer Select Bit	0: Not used 1: Used	RW
BRST	Burst Transfer Select Bit	0: Single transfer 1: Burst transfer	RW
INTE	DMA II Transfer Complete Interrupt Select Bit	0: Not used 1: Used	RW
CHAIN	Chained Transfer Select Bit	0: Not used 1: Used	RW
— (b14-b8)	Reserved	Should be written with 0	RW
MULT	Multiple Transfer Select Bit	0: Not used	RW

When multiple transfer is selected (MULT = 1)



Bit Symbol	Bit Name	Function	RW
SIZE	Transfer Size Select Bit	0: 8 bits 1: 16 bits	RW
IMM	Reserved	Should be written with 1	RW
UPDS	Source Addressing Select Bit	0: Non-incrementing addressing 1: Incrementing addressing	RW
UPDD	Destination Addressing Select Bit	0: Non-incrementing addressing 1: Incrementing addressing	RW
CNT0	Transfer Number Set Bit	b6 b5 b4 0 0 0 : Do not use this combination 0 0 1 : Once 0 1 0 : Twice : : 1 1 1 : Seven times	RW
CNT1			RW
CNT2			RW
CHAIN	Reserved	Should be written with 0	RW
— (b14-b8)	Reserved	Should be written with 0	RW
MULT	Multiple Transfer Select Bit	1: Used	RW

Note:
1. The MOD should be located on the RAM.

Figure 13.3 MOD

13.1.3 Interrupt Control Register of the Peripheral Function

Set bits ILVL2 to ILVL0 in the interrupt control register for the peripheral interrupt triggering DMAC II to 111b (level 7).

13.1.4 Relocatable Vector Table of the Peripheral Function

Set the starting address of the DMAC II index to the interrupt vector for the peripheral interrupt triggering DMAC II.
To use the chained transfer, locate the relocatable vector table on the RAM.

13.1.5 IRLT Bit in the IIOiIE Register (i = 0 to 11)

To use the intelligent I/O interrupt as a trigger for DMAC II, set the IRLT bit in the corresponding IIOiIE register to 0 (interrupt request for DMA or DMA II used).

13.2 DMAC II Performance

To perform a DMA II transfer, the DMAII bits in registers RIPL1 and RIPL2 should be set to 1 (interrupt request level 7 used for DMA II transfer). Any peripheral interrupts with bits ILVL2 to ILVL0 set to 111b (level 7) can be a request source to activate DMAC II. These peripheral interrupt requests are available only for DMA II transfer, that is, they cannot be used for CPU.
When an interrupt request is generated with interrupt request level 7, DMAC II is activated irrespective of the state of I flag or IPL.
When a peripheral interrupt request triggering DMAC II and a higher-priority request such as watchdog timer interrupt, oscillator stop detection interrupt, and NMI are simultaneously generated, the higher-priority interrupt is accepted prior to the DMA II transfer, and the DMA II transfer starts after the higher-priority interrupt sequence.

13.3 Transfer Types

DMAC II transfers three types of 8-bit or 16-bit data as follows:

- Memory-to-memory transfer: Data is transferred from a given memory location in a 64-Mbyte space (addresses 00000000h to 01FFFFFFh and FE000000h to FFFFFFFFh) to another given memory location in the same space.
- Immediate data transfer: Immediate data is transferred to a given memory location in a 64-Mbyte space.
- Calculation transfer: Two data are added together and the result is transferred to a given memory location in a 64-Kbyte space.

When 16-bit data is transferred to DADR at FFFFFFFFh, it is transferred to 00000000h as well as FFFFFFFFh. The same transfer is performed when SADR is FFFFFFFFh.

13.3.1 Memory-to-memory Transfer

Data transfer between any two memory locations can be:

- A transfer from a constant address to another constant address
- A transfer from a constant address to an address range in memory
- A transfer from an address range in memory to a constant address
- A transfer from an address range in memory to another address range in memory

When increment addressing mode is selected, SADR and DADR increment by one in a 8-bit transfer and by two in a 16-bit transfer after a data transfer for the next transfer. When SADR or DADR exceeds FFFFFFFFh as a result of address incrementation, it returns to 00000000h. Likewise, when SADR or DADR exceeds 01FFFFFFh, it must become 02000000h, but an actual transfer is performed for FE000000h.

13.3.2 Immediate Data Transfer

DMAC II transfers immediate data to any memory location. Both incrementing or non-incrementing addressing modes are available for destination address. Store the immediate data to be transferred into SADR. To transfer 8-bit immediate data, set the data to the one lower byte of SADR. For 16-bit immediate data, set the data to the two lower bytes. The three upper bytes or the two upper bytes of respective case are ignored.

13.3.3 Calculation Transfer

After two memory data or an immediate data and memory data are added together, DMAC II transfers calculated result to any memory location. Set one address to be calculated or an immediate data to SADR and set the other address to be calculated to OADR. Both incrementing or non-incrementing addressing modes are available for source and destination addresses in case of a data in memory + a data in memory calculation transfer. If the source addressing is incrementing mode, the operation addressing should be also incrementing. In case of an immediate data + a data in memory calculation transfer, the addressing mode is selectable only for destination address.

13.4 Transfer Modes

DMAC II provides three types of basic transfer modes: single transfer, burst transfer, and multiple transfer. COUNT determines the number of transfers to be performed. No transfer is performed when COUNT is set to 0000h.

13.4.1 Single Transfer

Set the BRST bit in the MOD to 0.
One data transfer is performed by one transfer request.
When incrementing addressing mode is selected for the source and/or destination address, the address(es) increment(s) after a data transfer for the next transfer.
COUNT is decremented every time a data transfer is performed. When COUNT reaches 0000h, the DMA II transfer complete interrupt request is generated if the INTE bit in the MOD is 1 (the DMA II transfer complete interrupt used).

13.4.2 Burst Transfer

Set the BRST bit in the MOD to 1.
DMAC II continuously transfers data for the number of times determined by COUNT by one transfer request. COUNT is decremented every time a data transfer is performed. When COUNT reaches 0000h, the burst transfer is completed. The DMA II transfer complete interrupt request is generated if the INTE bit is 1 (the DMA II transfer complete interrupt used).
No interrupt is accepted during burst transfer being performed.

13.4.3 Multiple Transfer

Set the MULT bit in the MOD to 1.
Multiple memory-to-memory transfers are performed from different source addresses to different destination addresses by one transfer request.
Bits CNT2 to CNT0 in the MOD select the number of transfers to be performed from 001b (once) to 111b (seven times). These bits should not be set to 000b.
Allocate required number of SDARs and DADRs alternately following MOD and COUNT.
When the multiple transfer is selected, the following transfer functions are not available: the calculation transfer, burst transfer, chained transfer, and DMA II transfer complete interrupt.

13.5 Chained Transfer

The chained transfer is available when the CHAIN bit in the MOD is set to1.
The chained transfer is performed as follows:

(1) When a transfer request is generated, a data transfer is performed according to DMAC II index specified by the corresponding interrupt vector. Either single transfer (the BRST bit in the MOD is 0) or burst transfer (the BRST bit is 1) is performed according to the BRST bit setting.
(2) When COUNT reaches 0000h, the value in the interrupt vector in (1) above is overwritten with the value in CADR. Simultaneously, the DMA II transfer complete interrupt is generated when the INTE bit in the MOD is 1.
(3) When the next DMA II transfer request is generated, the data transfer is performed according to DMAC II index specified by the peripheral interrupt vector in (2) above.

Figure 13.4 shows the relocatable vector and DMAC II index in chained transfer.
To use the chained transfer, the relocatable vector table should be located on the RAM.

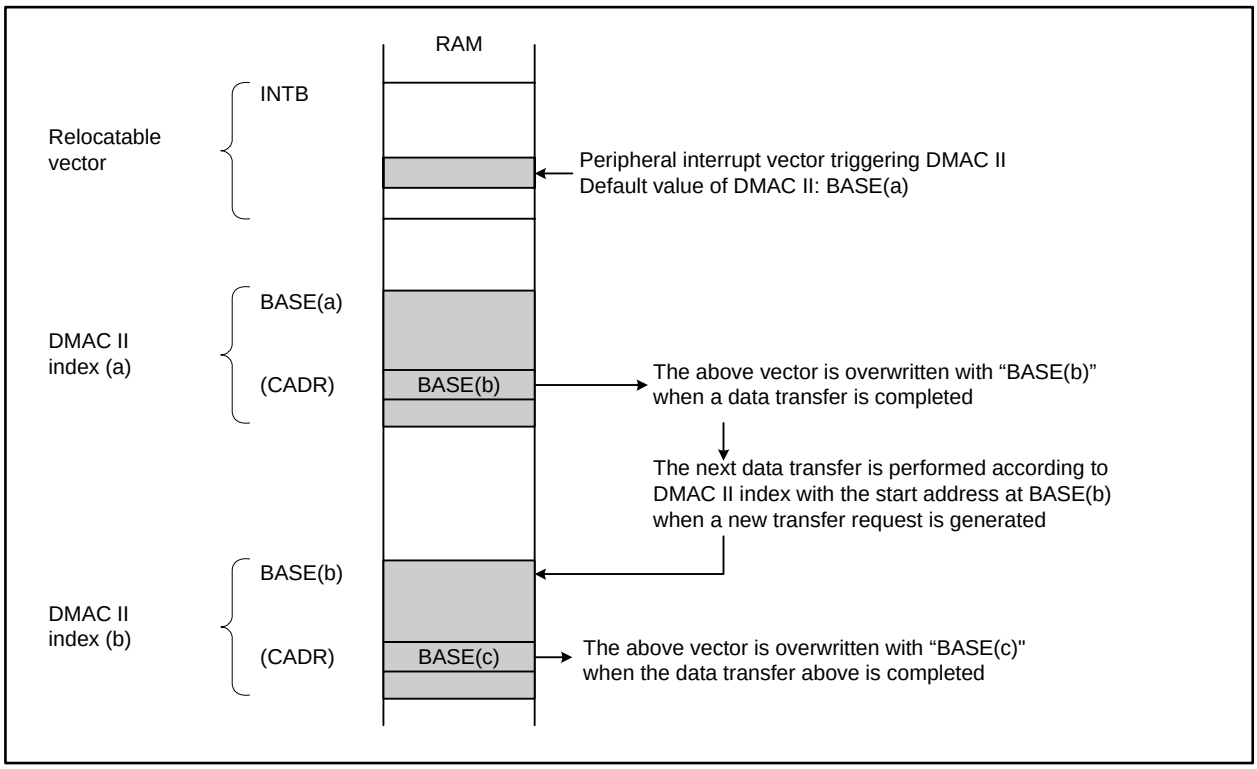

RAM
INTB
Relocatable vector
Peripheral interrupt vector triggering DMAC II
Default value of DMAC II: BASE(a)
BASE(a)
DMAC II index (a)
(CADR)
BASE(b)
The above vector is overwritten with “BASE(b)” when a data transfer is completed
The next data transfer is performed according to DMAC II index with the start address at BASE(b) when a new transfer request is generated
BASE(b)
DMAC II index (b)
(CADR)
BASE(c)
The above vector is overwritten with “BASE(c)" when the data transfer above is completed

Figure 13.4 Relocatable Vector and DMAC II Index in Chained Transfer

13.6 DMA II Transfer Complete Interrupt

The DMA II transfer complete interrupt is available when the INTE bit in the MOD is set to1.
The starting address of the DMA II transfer complete interrupt handler should be set to IADR. The interrupt is generated when COUNT reaches 0000h.

The initial instruction of the interrupt handler is executed in the eighth cycle after a DMA II transfer is completed.

13.7 Execution Time

DMAC II execution cycle is calculated by the following equations:

Other than multiple transfer: t = 6 + (26 + a + b + c + d) × m + (4 + e) × n cycles
Multiple transfer: t = 21 + (11 + b + c) × k cycles

a: if IMM = 0 (transfer source is immediate data), a = 0;
if IMM = 1 (transfer source is memory), a = -1
b: if UPDS = 1 (source addressing is incrementing), b = 0;
if UPDS = 0 (source addressing is non-incrementing), b = 1
c: if UPDD = 1 (destination addressing is incrementing), c = 0;
if UPDD = 0 (destination addressing is non-incrementing), c = 1
d: if OPER = 0 (calculation transfer is not selected), d = 0;
if OPER = 1 (calculation transfer is selected) and UPDS = 0 (source addressing is immediate data or non-incrementing),
d = 7;
if OPER = 1 (calculation transfer is selected) and UPDS = 1 (source addressing is incrementing),
d = 8
e: if CHAIN = 0 (chained transfer is not selected), e = 0;
if CHAIN = 1 (chained transfer is selected), e = 4
m: if BRST = 0 (single transfer), m = 1;
if BRST = 1 (burst transfer), m = COUNT
n: if COUNT = 0001h, n = 0; if COUNT= 0002h or more, n = 1
k: The number of transfers to be performed set using bits CNT2 to CNT0

The equations above are approximate. The cycles may vary depending on CPU state, bus wait state and DMAC II index allocation.

When a DMA II transfer complete interrupt (transfer counter = 2) is generated with no chained transfer after a memory-to-memory transfer is performed twice with a incremented source address and a non-incremented destination address in single transfer mode

(a = -1, b = 0, c = 1, d = 0, e = 0, m = 1)

First DMA II transfer t = 6 + (26 - 1 + 0 + 1 + 0) × 1 + (4 + 0) × 1 = 36 cycles
Second DMA II transfer t = 6 + (26 - 1 + 0 + 1 + 0) × 1 + (4 + 0) × 0 = 32 cycles

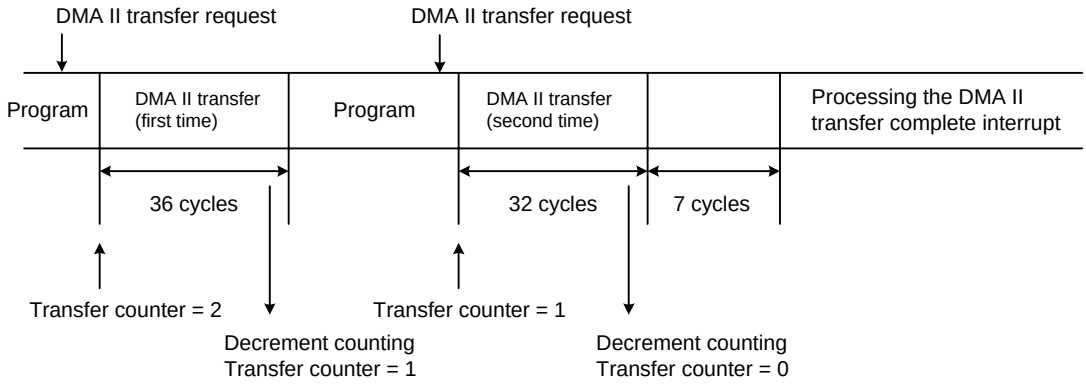


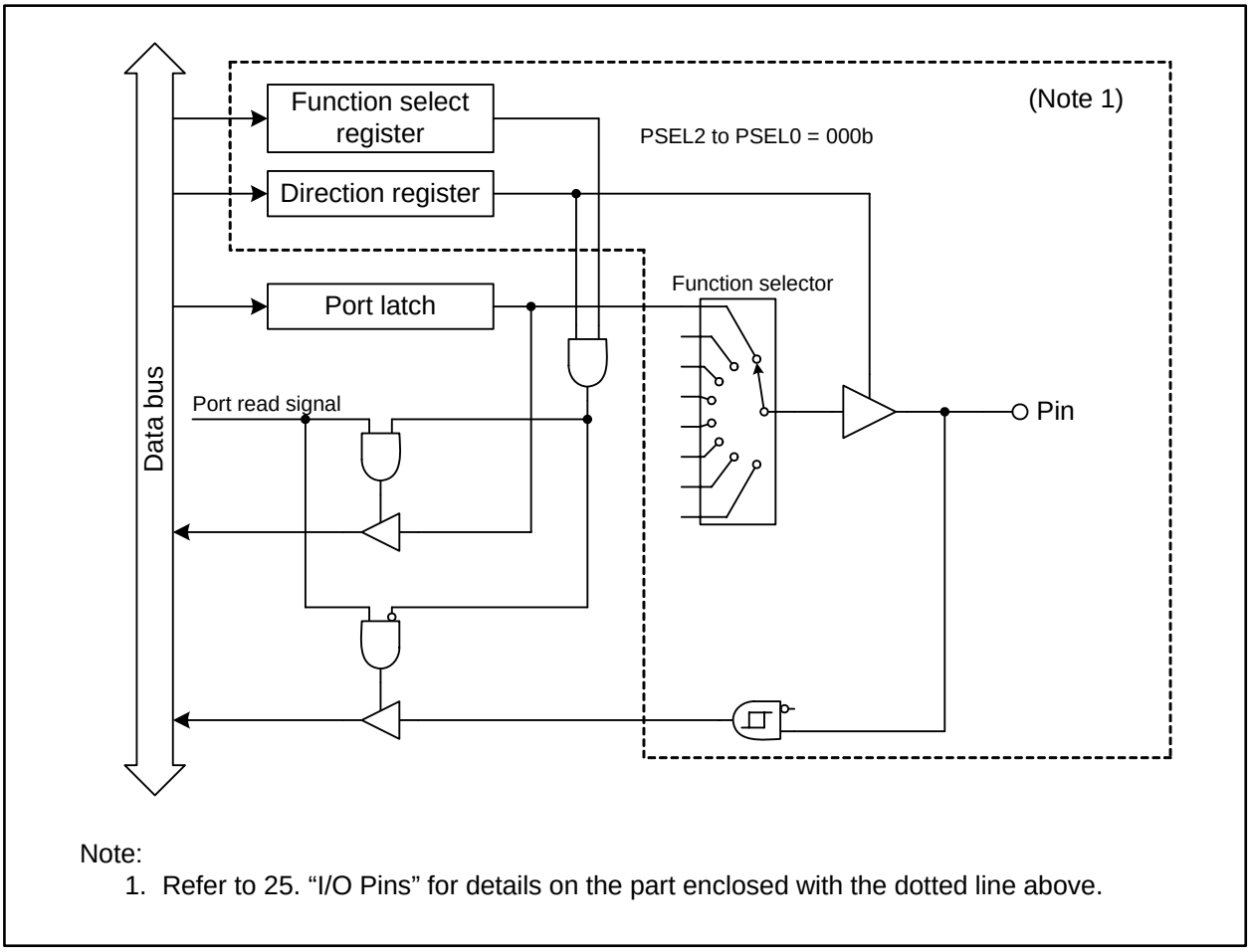
Figure 13.5 Transfer Cycles

14. Programmable I/O Ports

The programmable I/O ports are designated as 113 ports from P0 to P15 (excluding P8_5 and P14_0 to P14_2. P11_0 to P11_2 and P15_4 to P15_7 are also excluded since these ports are internally connected to CODEC).

Each port status, input or output, can be selected using the direction register except P8_5 and P14_1 which are input only. The P8_5 bit in the P8 register indicates an $\overline{\text{NMI}}$ input level since the P8_5 shares a pin with the $\overline{\text{NMI}}$.

Figure 14.1 shows a configuration of programmable I/O ports and Figure 14.2 and Figure 14.3 show a configuration of each input-only port.



Note:
1. Refer to 25. “I/O Pins” for details on the part enclosed with the dotted line above.

Figure 14.1 Programmable I/O Port Configuration

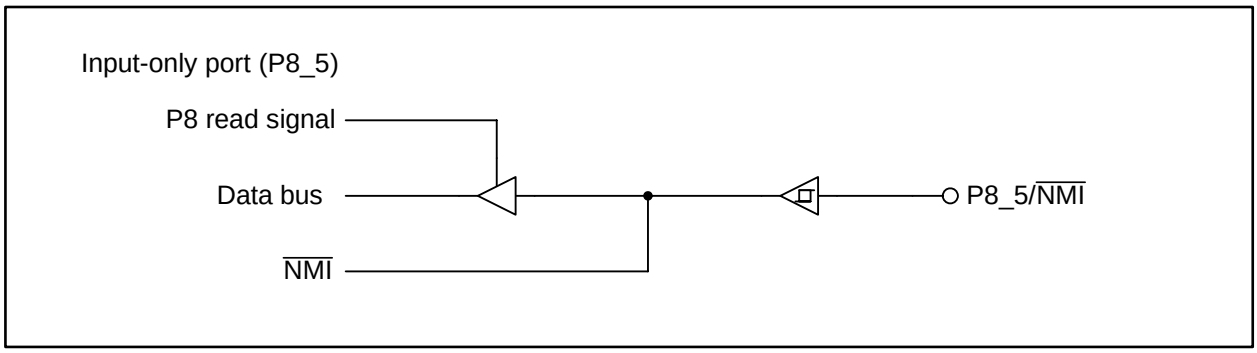


Figure 14.2 Input-only Port Configuration (1/2)

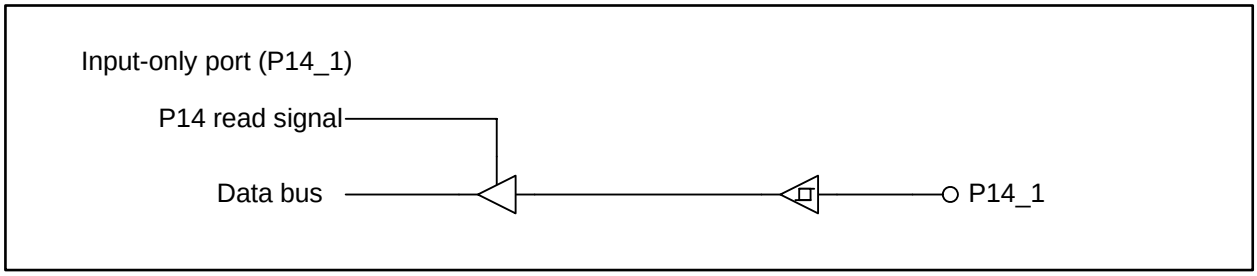


Figure 14.3 Input-only Port Configuration (2/2)

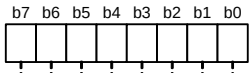
14.1 Port Pi Register (Pi register, i = 0 to 15)

A write/read to the Pi register is required to communicate with external devices. This register consists of a port latch to hold output data and a circuit to read pin states. Each bit in the Pi register corresponds to a respective port.

When a programmable I/O port is selected in the output function select register, the value in the port latch as output data and pin states as input data are respectively read.

Figure 14.4 shows the Pi register.

Port Pi Register (i = 0 to 15) (1)



Symbol	Address	Reset Value
P0 to P3	03C0h, 03C1h, 03C4h, 03C5h	Undefined
P4 to P7	03C8h, 03C9h, 03CCh, 03CDh	Undefined
P8 (2), P9	03D0h, 03D1h	Undefined
P10, P11 (1, 4)	03D4h, 03D5h	Undefined
P12, P13	03D8h, 03D9h	Undefined
P14 (2, 3, 4)	03DCh	Undefined
P15 (1)	03DDh	Undefined

Bit Symbol	Bit Name	Function	RW
Pi_0	Port Pi_0 bit (4)	When the direction bit = 0 (input) A value is written to the corresponding bit. It is not output due to input mode selected. The read value is the corresponding pin state as follows: 0: Low 1: High When the direction bit = 1 (output) A value is written to the corresponding bit as follows: 0: Output low 1: Output high The read value has the same output level as that written to the corresponding bit	RW
Pi_1	Port Pi_1 bit (3)		RW
Pi_2	Port Pi_2 bit (4)		RW
Pi_3	Port Pi_3 bit		RW
Pi_4	Port Pi_4 bit		RW
Pi_5	Port Pi_5 bit (3, 5)		RW
Pi_6	Port Pi_6 bit (5)		RW
Pi_7	Port Pi_7 bit (5)		RW

Notes:
1. Ports P11_0 to P11_2 and P15_4 to P15_7 are internally connected to CODEC. No pins are provided for these ports.
2. The P8_5 bit in the P8 register and the P14_1 bit in the P14 register are read only.
3. Bits P14_0 and P14_2 in the P14 register are reserved. These bits should be written with 0 and read as undefined values.
4. Bits P11_5 to P11_7 in the P11 register and the P14_7 bit in the P14 register are unavailable on this MCU. These bits should be written with 0 and read as undefined values.

Figure 14.4 Registers P0 to P15

15. Timers

This MCU has eleven 16-bit timers which are divided into two groups according to functions: five timer As and six timer Bs. Each timer functions individually. The count source of each timer provides the clock for timer operations including counting, reloading and so on.
Figure 15.1 and Figure 15.2 show the configuration of the timers A and B, respectively.

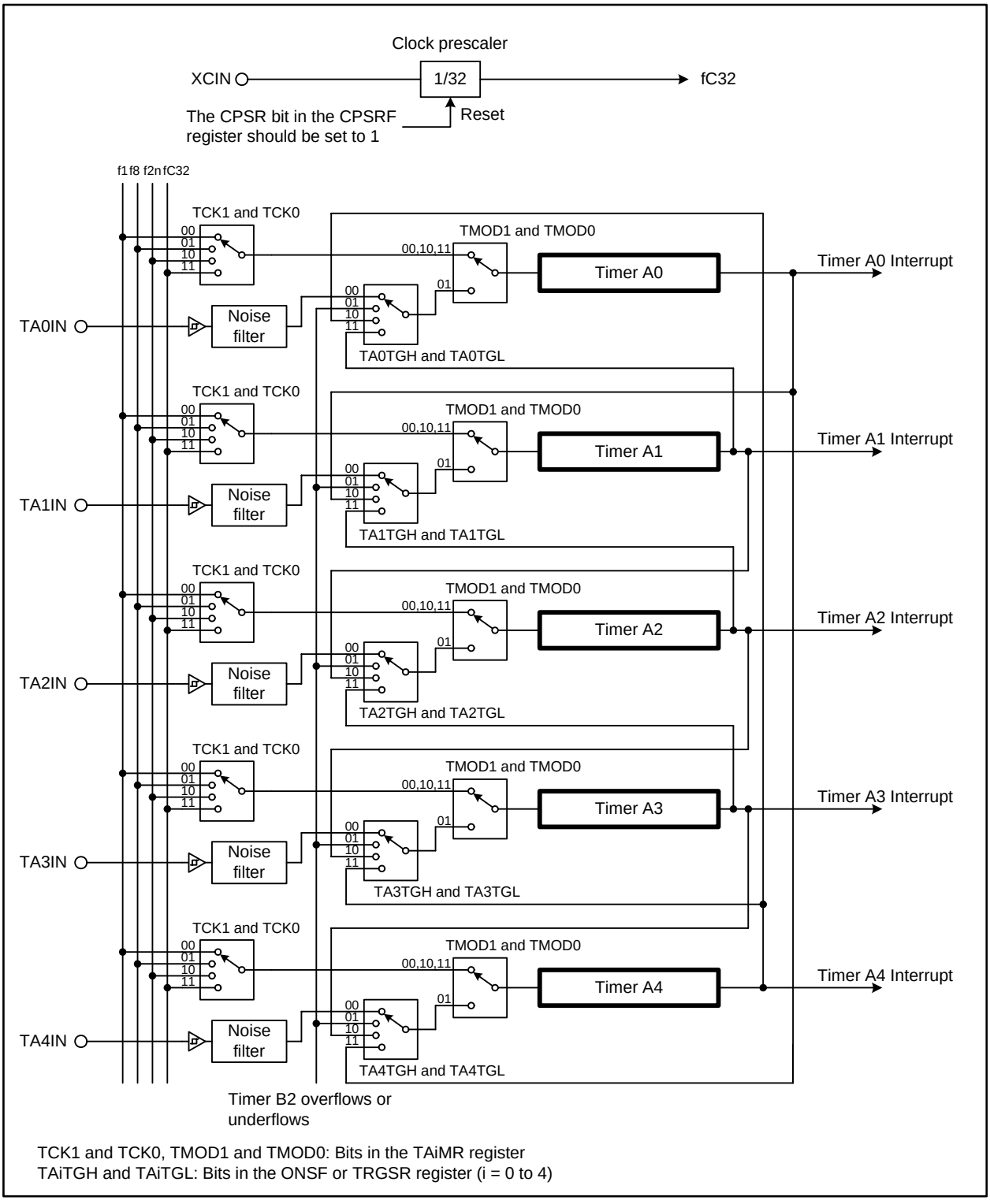


Figure 15.1 Timer A Configuration

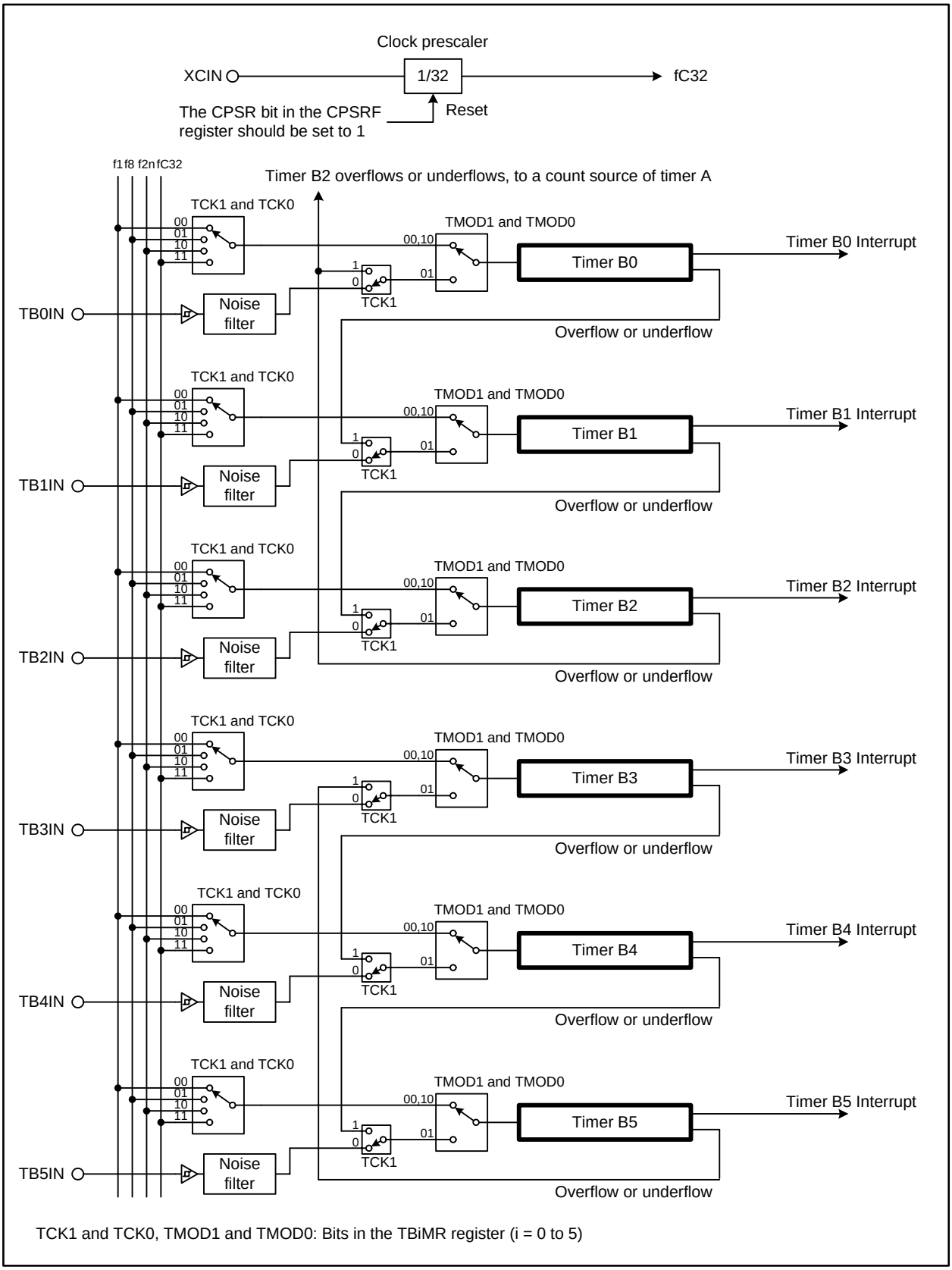


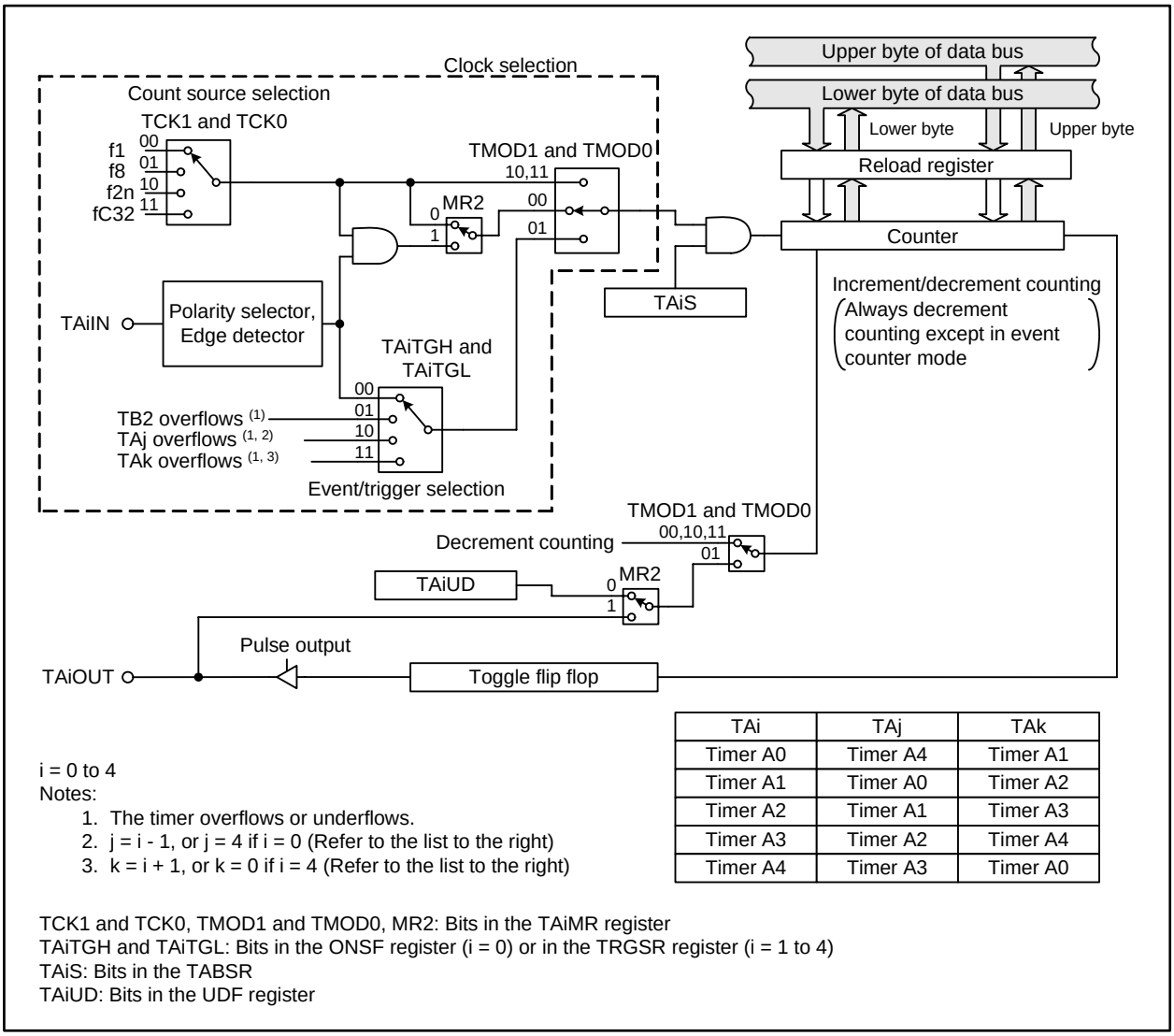
Figure 15.2 Timer B Configuration

15.1 Timer A

Figure 15.3 shows a block diagram of the timer A and Figure 15.4 to Figure 15.10 show registers associated with the timer A.

The timer A supports four modes shown as below. Timers A0 to A4 in any mode other than the event counter mode have the same function. A mode is selected using bits TMOD1 and TMOD0 in the TAiMR register (i = 0 to 4).

- Timer mode: The timer counts an internal count source
- Event counter mode: The timer counts an external pulse or an overflow and underflow of other timers
- One-shot timer mode: The timer outputs one valid pulse before the counter reaches 0000h
- Pulse-width modulation mode: The timer sequentially outputs pulses of given width



TAi	TAj	TAk
Timer A0	Timer A4	Timer A1
Timer A1	Timer A0	Timer A2
Timer A2	Timer A1	Timer A3
Timer A3	Timer A2	Timer A4
Timer A4	Timer A3	Timer A0

i = 0 to 4

Notes:
1. The timer overflows or underflows.
2. j = i - 1, or j = 4 if i = 0 (Refer to the list to the right)
3. k = i + 1, or k = 0 if i = 4 (Refer to the list to the right)

TCK1 and TCK0, TMOD1 and TMOD0, MR2: Bits in the TAiMR register
TAiTGH and TAiTGL: Bits in the ONSF register (i = 0) or in the TRGSR register (i = 1 to 4)
TAiS: Bits in the TABSR
TAiUD: Bits in the UDF register

Figure 15.3 Timer A Block Diagram

Timer Ai Register (i = 0 to 4) (1)

b15 b8 b7 b0

Symbol	Address	Reset Value
TA0 to TA2	0347h-0346h, 0349h-0348h, 034Bh-034Ah	Undefined
TA3, TA4	034Dh-034Ch, 034Fh-034Eh	Undefined

Mode	Function	Setting Range	RW
Timer Mode	Divides the count source by n+1 (n = setting value)	0000h to FFFFh	RW
Event Counter Mode	Divides the count source by FFFFh -n+1 (on increment counting) or by n+1 (on decrement counting) (n = setting value) (2)	0000h to FFFFh	RW
One-shot Timer Mode	Divides the count source by n, then stops (n = setting value) (3)	0000h to FFFFh (4)	WO
Pulse-width Modulation Mode (16-bit PWM)	PWM period: (2^{16} -1) / fj High level width of PWM pulse: n / fj (fj = count source frequency, n = setting value of the TAi register) (5)	0000h to FFFEh (4)	WO
Pulse-width Modulation Mode (8-bit PWM)	PWM period: (2^{8} -1) × (m+1) / fj High level width of PWM pulse: (m+1)n / fj (fj = count source frequency, n = setting value of the upper byte in the TAi register, m = setting value of the lower byte in the TAi register) (5)	00h to FEh (upper byte) 00h to FFh (lower byte) (4)	WO

fj: f1, f8, f2n, fC32

Notes:
1. A 16-bit read/write access to this register should be performed.
2. The timer counts an external input pulse or an overflow and underflow of other timers.
3. When the TAi register is set to 0000h, the timer counter does not start, nor the TAi interrupt request is generated.
4. The MOV instruction should be used to set the TAi register.
5. When the TAi register is set to 0000h, the pulse-width modulator does not operate, the TAiOUT pin is held low, and the TAi interrupt request is not generated. The same situation occurs in 8-bit pulse-width modulator mode if the upper byte in the TAi register are set to 00h.

Figure 15.4 Registers TA0 to TA4

Timer Ai Mode Register (i = 0 to 4)

b7	b6	b5	b4	b3	b2	b1	b0
					0		

Symbol	Address	Reset Value
TA0MR to TA4MR	0356h, 0357h, 0358h, 0359h, 035Ah	0000 0000b

Bit Symbol	Bit Name	Function	RW
TMOD0	Operating Mode Select Bit	b1 b0 0 0 : Timer mode 0 1 : Event counter mode 1 0 : One-shot timer mode 1 1 : Pulse-width modulation mode	RW
TMOD1			RW
— (b2)	Reserved	Should be written with 0	RW
MR1	—	Function varies according to the operating mode	RW
MR2			RW
MR3			RW
TCK0	Count Source Select Bit	Function varies according to the operating mode	RW
TCK1			RW

Figure 15.5 Registers TA0MR to TA4MR

Count Start Register

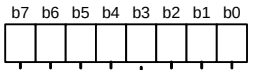
b7	b6	b5	b4	b3	b2	b1	b0

Symbol	Address	Reset Value
TABSR	0340h	0000 0000b

Bit Symbol	Bit Name	Function	RW
TA0S	Timer A0 Count Start Bit	0: Stop counter 1: Start counter	RW
TA1S	Timer A1 Count Start Bit	0: Stop counter 1: Start counter	RW
TA2S	Timer A2 Count Start Bit	0: Stop counter 1: Start counter	RW
TA3S	Timer A3 Count Start Bit	0: Stop counter 1: Start counter	RW
TA4S	Timer A4 Count Start Bit	0: Stop counter 1: Start counter	RW
TB0S	Timer B0 Count Start Bit	0: Stop counter 1: Start counter	RW
TB1S	Timer B1 Count Start Bit	0: Stop counter 1: Start counter	RW
TB2S	Timer B2 Count Start Bit	0: Stop counter 1: Start counter	RW

Figure 15.6 TABSR Register

Increment/Decrement Counting Select Register (1)



Symbol	Address	Reset Value
UDF	0344h	0000 0000b

Bit Symbol	Bit Name	Function	RW
TA0UD	Timer A0 Increment/Decrement Counting Select Bit	0: Count decremented 1: Count incremented (2)	RW
TA1UD	Timer A1 Increment/Decrement Counting Select Bit	0: Count decremented 1: Count incremented (2)	RW
TA2UD	Timer A2 Increment/Decrement Counting Select Bit	0: Count decremented 1: Count incremented (2)	RW
TA3UD	Timer A3 Increment/Decrement Counting Select Bit	0: Count decremented 1: Count incremented (2)	RW
TA4UD	Timer A4 Increment/Decrement Counting Select Bit	0: Count decremented 1: Count incremented (2)	RW
TA2P	Timer A2 Two-phase Pulse Signal Processing Select Bit	0: Two-phase pulse signal processing disabled 1: Two-phase pulse signal processing enabled (3)	WO
TA3P	Timer A3 Two-phase Pulse Signal Processing Select Bit	0: Two-phase pulse signal processing disabled 1: Two-phase pulse signal processing enabled (3)	WO
TA4P	Timer A4 Two-phase Pulse Signal Processing Select Bit	0: Two-phase pulse signal processing disabled 1: Two-phase pulse signal processing enabled (3)	WO

Notes:
1. The MOV instruction should be used to set this register.
2. This bit is enabled when the MR2 bit in the TAiMR register (i = 0 to 4) is set to 0 (the UDF register setting is the source of increment/decrement count switching) in event counter mode.
3. This bit should be set to 0 when the two-pulse signal processing is not in use.

Figure 15.7 UDF Register

One-shot Start Register

b7	b6	b5	b4	b3	b2	b1	b0

Symbol	Address	Reset Value
ONSF	0342h	0000 0000b

Bit Symbol	Bit Name	Function	RW
TA0OS	Timer A0 One-shot Start Bit	0: Timer in idle state 1: Start the timer (1)	RW
TA1OS	Timer A1 One-shot Start Bit	0: Timer in idle state 1: Start the timer (1)	RW
TA2OS	Timer A2 One-shot Start Bit	0: Timer in idle state 1: Start the timer (1)	RW
TA3OS	Timer A3 One-shot Start Bit	0: Timer in idle state 1: Start the timer (1)	RW
TA4OS	Timer A4 One-shot Start Bit	0: Timer in idle state 1: Start the timer (1)	RW
TAZIE	Z-phase Input Enable Bit	0: Z-phase input disabled 1: Z-phase input enabled	RW
TA0TGL	Timer A0 Event/Trigger Select Bit	b7 b6 0 0 : Select the input to the TA0IN pin 0 1 : Select the overflow of TB2 (2) 1 0 : Select the overflow of TA4 (2) 1 1 : Select the overflow of TA1 (2)	RW
TA0TGH			RW

Notes:
1. This bit is read as 0.
2. The timer overflows or underflows.

Figure 15.8 ONSF Register

Trigger Select Register

b7	b6	b5	b4	b3	b2	b1	b0

Symbol	Address	Reset Value
TRGSR	0343h	0000 0000b

Bit Symbol	Bit Name	Function	RW
TA1TGL	Timer A1 Event/Trigger Select Bit	b1 b0 0 0 : Select the input to the TA1IN pin 0 1 : Select the overflow of TB2 (1) 1 0 : Select the overflow of TA0 (1) 1 1 : Select the overflow of TA2 (1)	RW
TA1TGH			RW
TA2TGL	Timer A2 Event/Trigger Select Bit	b2 b3 0 0 : Select the input to the TA2IN pin 0 1 : Select the overflow of TB2 (1) 1 0 : Select the overflow of TA1 (1) 1 1 : Select the overflow of TA3 (1)	RW
TA2TGH			RW
TA3TGL	Timer A3 Event/Trigger Select Bit	b4 b5 0 0 : Select the input to the TA3IN pin 0 1 : Select the overflow of TB2 (1) 1 0 : Select the overflow of TA2 (1) 1 1 : Select the overflow of TA4 (1)	RW
TA3TGH			RW
TA4TGL	Timer A4 Event/Trigger Select Bit	b6 b7 0 0 : Select the input to the TA4IN pin 0 1 : Select the overflow of TB2 (1) 1 0 : Select the overflow of TA3 (1) 1 1 : Select the overflow of TA0 (1)	RW
TA4TGH			RW

Note:
1. The timer overflows or underflows.

Figure 15.9 TRGSR Register

Count Source Prescaler Register

b7	b6	b5	b4	b3	b2	b1	b0
	0	0	0				

Symbol	Address	Reset Value
TCSPR	035Fh	0000 0000b

Bit Symbol	Bit Name	Function	RW
CNT0	Divide Ratio Select Bit (1)	f2n is either the main clock or peripheral clock source divided by 2n. If n = 0, the clock is not divided (n = setting value)	RW
CNT1			RW
CNT2			RW
CNT3			RW
— (b6-b4)	Reserved	Should be written with 0	RW
CST	Divider Operation Enable Bit	0: Stop divider operation 1: Start divider operation	RW

Note:
1. The CST bit should be set to 0 to rewrite bits CNT3 to CNT0.

Figure 15.10 TCSPR Register

15.1.1 Timer Mode

In timer mode, the timer counts an internally generated count source. Table 15.1 lists specifications of timer mode. Figure 15.11 shows registers TA0MR to TA4MR in this mode.

Table 15.1 Timer Mode Specifications (i = 0 to 4)

Item	Specification
Count sources	f1, f8, f2n, or fC32
Count operations	• Decrement counting • If the timer counter underflows, the reload register setting is reloaded into the counter to resume counting
Divide ratio	$\frac{1}{n+1}$ n: TAi register setting value, 0000h to FFFFh
Count start condition	The TAiS bit in the TABSR register is set to 1 (count starts)
Count stop condition	The TAiS bit in the TABSR register is set to 0 (count stops)
Interrupt request generating timing	When the timer counter underflows
TAiIN pin function	A programmable I/O port or a gate input
TAiOUT pin function	A programmable I/O port or a pulse output
Read from timer	The TAi register indicates a counter value
Write to timer	• While the timer counter is stopped or before the initial count source is input after starting to count, the value written to the TAi register is written to both reload register and the counter • While the timer counter is running, the value written to the TAi register is written to the reload register (It is transferred to the counter at the next reload timing)
Selectable functions	• Gate function Input signal to the TAiIN pin can control to start/stop counting • Pulse output function The polarity of the TAiOUT pin is inverted whenever the timer counter underflows. A low is output while the TAiS bit holds 0 (count stops)

Timer Ai Mode Register (i = 0 to 4) (timer mode)

b7	b6	b5	b4	b3	b2	b1	b0
		0			0	0	0

Symbol	Address	Reset Value
TA0MR to TA4MR	0356h, 0357h, 0358h, 0359h, 035Ah	0000 0000b

Bit Symbol	Bit Name	Function	RW
TMOD0	Operating Mode Select Bit	b1 b0 0 0 : Timer mode	RW
TMOD1			RW
— (b2)	Reserved	Should be written with 0	RW
MR1	Gate Function Select Bit	b4 b3 0 X : No gate function (1) (TAiIN pin functions as programmable I/O port) 1 0 : Count only while the TAiIN pin is held low 1 1 : Count only while the TAiIN pin is held high	RW
MR2			RW
MR3	Should be written with 0 in timer mode		RW
TCK0	Count Source Select Bit	b7 b6 0 0 : f1 0 1 : f8 1 0 : f2n 1 1 : fC32	RW
TCK1			RW

Note:
1. X can be set to either 0 or 1.

Figure 15.11 Registers TA0MR to TA4MR in Timer Mode

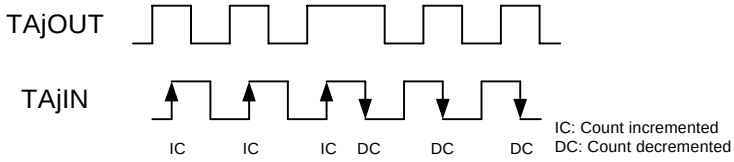
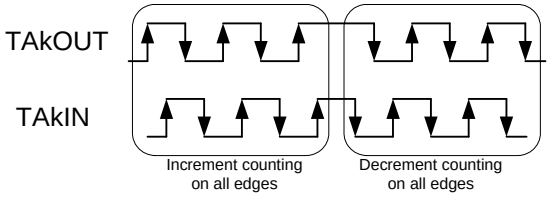
15.1.2 Event Counter Mode

In event counter mode, the timer counts an external signal or an overflow and underflow of other timers. Timers A2, A3 and A4 can count two-phase external signals. Table 15.2 lists specification in event count mode and Table 15.3 also list the specification when the timers use two-phase pulse signal processing. Figure 15.12 shows registers TA0MR to TA4MR in this mode.

Table 15.2 Event Counter Mode Specifications (without two-phase pulse signal processing) (i = 0 to 4)

Item	Specification
Count sources	• External signal applied to the TAiIN pin (valid edge is selectable by a program) • The overflow or underflow signal of timer B2, timer Aj (j = i - 1, or j = 4 if i = 0), and timer Ak (k = i + 1, or k = 0 if i = 4)
Count operations	• Increment/decrement counting can be switched by an external signal or program • If the timer counter underflows or overflows, the reload register setting is reloaded into the counter to resume counting. In the free-running count operation, the timer counter continues counting without reloading
Divide ratio	• $\frac{1}{FFFFh - n + 1}$ for increment counting • $\frac{1}{n + 1}$ for decrement counting n: TAi register setting value, 0000h to FFFFh
Count start condition	The TAiS bit in the TABSR register is set to 1 (count starts)
Count stop condition	The TAiS bit in the TABSR register is set to 0 (count stops)
Interrupt request generating timing	When the timer counter overflows or underflows
TAiIN pin function	A programmable I/O port or a count source input
TAiOUT pin function	A programmable I/O port, a pulse output, or an input for increment/decrement count switching
Read from timer	The TAi register indicates a counter value
Write to timer	• While the timer counter is stopped or before the initial count source is input after starting to count, the value written to the TAi register is written to both reload register and the counter • While the timer counter is running, the value written to the TAi register is written to the reload register (It is transferred to the counter at the next reload timing)
Selectable functions	• Free-running count function The reload register setting is not reloaded even if the timer counter overflows or underflows • Pulse output function The polarity of the TAiOUT pin is inverted whenever the timer counter overflows or underflows. A low is output while the TAiS bit holds 0 (count stops)

Table 15.3 Event Counter Mode Specifications (with two-phase pulse signal processing on timers A2 to A4) (i = 2 to 4)

Item	Specification
Count sources	Two-phase pulse signal applied to pins TAiIN and TAiOUT
Count operations	• Increment/decrement counting can be switched by a two-phase pulse signal • If the timer counter underflows or overflows, the reload register setting is reloaded into the counter to resume counting. In the free-running count operation, the timer counter continues counting without reloading
Divide ratio	• $\frac{1}{FFFFh - n + 1}$ for increment counting • $\frac{1}{n + 1}$ for decrement counting n: TAi register setting value, 0000h to FFFFh
Count start condition	The TAiS bit in the TABSR register is set to 1 (count starts)
Count stop condition	The TAiS bit in the TABSR register is set to 0 (count stops)
Interrupt request generating timing	When the timer counter overflows or underflows
TAiIN pin function	A two-phase pulse input
TAiOUT pin function	A two-phase pulse input
Read from timer	The TAi register indicates a counter value
Write to timer	• While the timer counter is stopped or before the initial count source is input after starting to count, the value written to the TAi register is written to both reload register and the counter • While the timer counter is running, the value written to the TAi register is written to the reload register (It is transferred to the counter at the next reload timing)
Selectable functions (1)	• Normal processing operation (timers A2 and A3) While the input signal applied to the TAjOUT pin (j = 2 or 3) is held high, the timer increments the count on the rising edge of the TAjIN pin and decrements the count on the falling edge  • Quadrupled processing operation (timers A3 and A4) When the input signal applied to the TAkOUT pin (k = 3 or 4) is held high on the rising edge of the TAkIN pin, the timer increments the count on both the rising and falling edges of pins TAkOUT and TAkIN. When the signal is held high on the falling edge of the TAkIN pin, the timer decrements the count on both the rising and falling edges of pins TAkOUT and TAkIN  • Counter reset by Z-phase input (the timer A3) The counter value is set to 0 by Z-phase input

Note:
1. Only the timer A3 is available for any selectable functions. The timer A2 is exclusively for the normal processing operation and the timer A4 is for the quadrupled processing operation, respectively.

Timer Ai Mode Register (i = 0 to 4) (event counter mode)

b7	b6	b5	b4	b3	b2	b1	b0
		0			0	0	1

Symbol	Address	Reset Value
TA0MR to TA4MR	0356h, 0357h, 0358h, 0359h, 035Ah	0000 0000b

Bit Symbol	Bit Name	Function (without two-phase pulse signal processing)	Function (with two-phase pulse signal processing)	RW
TMOD0	Operating Mode Select Bit	b1 b0 0 1 : Event counter mode (1)		RW
TMOD1				RW
— (b2)	Reserved	Should be written with 0		RW
MR1	Count Polarity Select Bit (2)	0: Count falling edges 1: Count rising edges	Should be written with 0	RW
MR2	Increment/Decrement Count Switching Source Select Bit	0: UDF register setting 1: Input signal to the TAiOUT pin (3)	Should be written with 1	RW
MR3	Should be written with 0 in event counter mode			RW
TCK0	Count Operation Type Select Bit	0: Reloading 1: Free-running		RW
TCK1	Two-phase Pulse Processing Operation Select Bit (4, 5)	Should be written with 0	0: Normal processing operation 1: Quadrupled processing operation	RW

Notes:
1. Bits TAiTGH and TAiTGL in the ONSF or TRGSR register select the count source in event counter mode.
2. This bit setting is enabled only when an external signal is counted.
3. The timer decrements the count when the input signal to the TAiOUT pin is held low and increments the count when the signal is held high.
4. The TCK1 bit is enabled only in the TA3MR register.
5. For two-phase pulse signal processing, the TAjP bit in the UDF register (j = 2 to 4) should be set to 1 (two-phase pulse signal processing enabled) and bits TAiTGH and TAiTGL should be set to 00b (the input to the TAjIN pin).

Figure 15.12 Registers TA0MR to TA4MR in Event Counter Mode

15.1.2.1 Counter Reset by Two-phase Pulse Signal Processing

A Z-phase input signal resets the timer counter when a two-phase pulse signal is being processed.

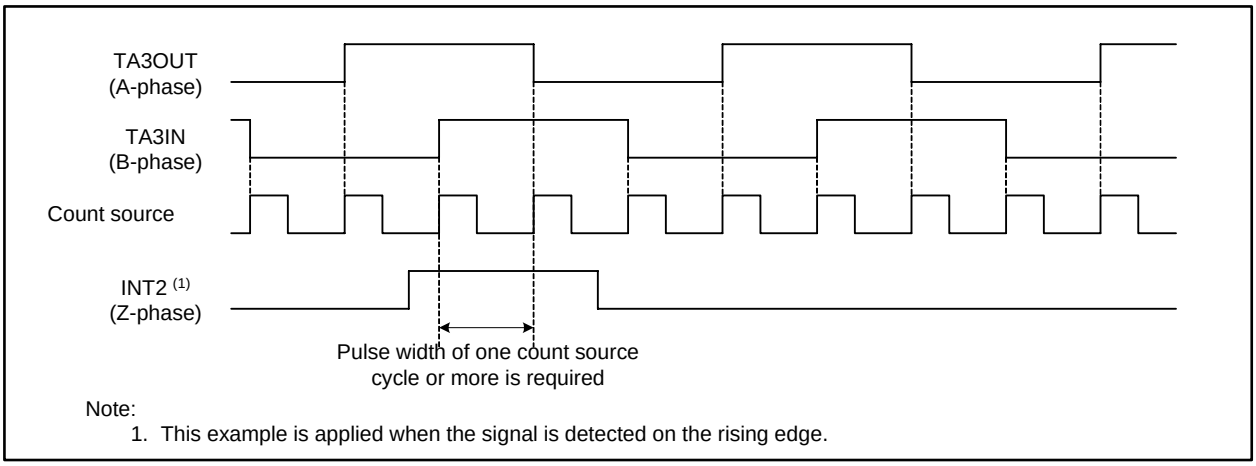
This function can be used under the following conditions: timer A3 event counter mode, two-phase pulse signal processing, free-running count operation type, and quadrupled processing. The Z-phase signal is applied to the $\overline{\text{INT2}}$ pin.

When the TAZIE bit in the ONSF register is set to 1 (Z-phase input enabled), the reset of timer counter by Z-phase input is enabled. To reset the counter, the TA3 register should be set to 0000h beforehand.

A Z-phase signal applied to the $\overline{\text{INT2}}$ pin is detected on a edge. The edge polarity is selected using the POL bit in the INT2IC register. The Z-phase signal should be input in order to have a pulse width of one count source cycle for timer A3 or more. Figure 15.13 shows the two-phase pulse (phases A and B) and the Z-phase.

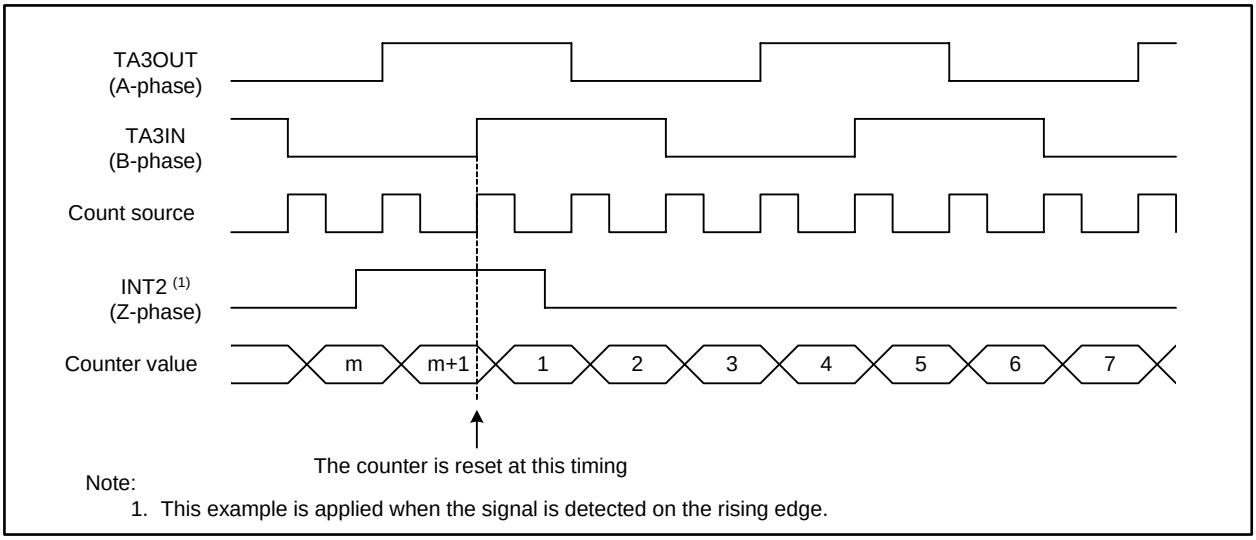
The timer counter is reset at the initial count source input after a Z-phase input is detected. Figure 15.14 shows the counter reset timing.

If the timer A3 overflows or underflows during a reset processing by the Z-phase input, two timer A3 interrupt requests are sequentially generated. To avoid this situation, the timer A3 interrupt request should not be used when this function is in use.



Note:
1. This example is applied when the signal is detected on the rising edge.

Figure 15.13 Two-phase Pulse (phases A and B) and Z-phase



Note:
1. This example is applied when the signal is detected on the rising edge.

Figure 15.14 Counter Reset Timing

15.1.3 One-shot Timer Mode

In one-shot timer mode, the timer operates only once for each trigger. Table 15.4 lists specifications of one-shot timer mode. Once a trigger occurs, the timer starts and operates for a given period. Figure 15.15 shows registers TA0MR to TA4MR in this mode.

Table 15.4 One-shot Timer Mode Specifications (i = 0 to 4)

Item	Specification
Count sources	f1, f8, f2n, or fC32
Count operations	• Decrement counting • When the timer counter reaches 0000h, it stops running after the reload register setting is reloaded • If a trigger occurs while counting, the reload register setting is reloaded into the counter to continue counting
Divide ratio	$\frac{1}{n}$ n: TAi register setting value, 0000h to FFFFh (Note that the timer counter does not run if n = 0000h)
Count start conditions	The TAiS bit in the TABSR register is set to 1 (count starts) and any of following triggers occurs: • An external trigger applied to the TAiIN pin • The overflow or underflow signal of timer B2, timer Aj (j = i - 1, or j = 4 if i = 0), or timer Ak (k = i + 1, or k = 0 if i = 4) • The TAiOS bit in the ONSF register is set to 1 (the timer started)
Count stop conditions	• The timer counter reaches 0000h and the reload register setting is reloaded • The TAiS bit in the TABSR register is set to 0 (count stops)
Interrupt request generating timing	When the timer counter reaches 0000h
TAiIN pin function	A programmable I/O port or a trigger input
TAiOUT pin function	A programmable I/O port or a pulse output
Read from timer	The TAi register indicates undefined value
Write to timer	• While the timer counter is stopped or before the initial count source is input after starting to count, the value written to the TAi register is written to both reload register and the counter • While the timer counter is running, the value written to the TAi register is written to the reload register (It is transferred to the counter at the next reload timing)
Selectable function	• Pulse output function A low is output while the timer counter is not running and a high is output while the timer counter is running

Timer Ai Mode Register (i = 0 to 4) (one-shot timer mode)

b7	b6	b5	b4	b3	b2	b1	b0
		0			0	1	0

Symbol	Address	Reset Value
TA0MR to TA4MR	0356h, 0357h, 0358h, 0359h, 035Ah	0000 0000b

Bit Symbol	Bit Name	Function	RW
TMOD0	Operating Mode Select Bit	b1 b0 1 0 : One-shot timer mode	RW
TMOD1			RW
— (b2)	Reserved	Should be written with 0	RW
MR1	External Trigger Select Bit (1)	0: Falling edge of input signal to the TAiIN pin 1: Rising edge of input signal to the TAiIN pin	RW
MR2	Trigger Select Bit	0: The TAiOS bit enabled 1: Selected using bits TAiTGH and TAiTGL	RW
MR3	Should be written with 0 in one-shot timer mode		RW
TCK0	Count Source Select Bit	b7 b6 0 0 : f1 0 1 : f8 1 0 : f2n 1 1 : fC32	RW
TCK1			RW

Note:
1. The MR1 bit setting is enabled only when bits TAiTGH and TAiTGL in the TRGSR register are set to 00b (the input to the TAiIN pin). This bit can be set to either 0 or 1 when bits TAiTGH and TAiTGL are set to 01b (the overflow or underflow of TB2), 10b (the overflow or underflow of TAi), or 11b (the overflow or underflow of TAi).

Figure 15.15 Registers TA0MR to TA4MR in One-shot Timer Mode

15.1.4 Pulse-width Modulation Mode

In pulse-width modulation mode, the timer outputs pulses of given width sequentially. Table 15.5 lists specifications of pulse-width modulation mode. The timer counter functions as either 16-bit or 8-bit pulse-width modulator. Figure 15.16 shows registers TA0MR to TA4MR in this mode. Figure 15.17 and Figure 15.18 respectively show an operation example of 16-bit and 8-bit pulse-width modulators.

Table 15.5 Pulse-width Modulation Mode Specification (i = 0 to 4)

Item	Specification
Count sources	f1, f8, f2n, or fC32
Count operations	• Decrement counting (the timer counter functions as an 8-bit or a 16-bit pulse-width modulator) • The reload register setting is reloaded on the rising edge of PWM pulse to resume counting • The timer is not affected by a trigger that is generated while the counter is running
16-bit PWM	• High level width: $\frac{n}{fj}$ n: TAi register setting value, 0000h to FFFEh fj: Count source frequency • Cycle: $\frac{2^{16}-1}{fj}$ fixed
8-bit PWM	• High level width: $\frac{n \times (m+1)}{fj}$ • Cycle: $\frac{(2^8-1) \times (m+1)}{fj}$ n: TAi register (upper byte) setting value, 00h to FEh m: TAi register (lower byte) setting value, 00h to FFh
Count start conditions	• The TAiS bit in the TABSR register is set to 1 (count starts) • The TAiS bit is set to 1 and an external trigger applied to the TAiIN pin • The TAiS bit is set to 1 and any of following triggers occurs: The overflow or underflow signal of timer B2, timer Aj (j = i - 1, or j = 4 if i = 0), or timer Ak (k = i + 1, or k = 0 if i = 4)
Count stop condition	The TAiS bit in the TABSR register is set to 0 (count stops)
Interrupt request generating timing	On the falling edge of the PWM pulse
TAiIN pin function	A programmable I/O port or a trigger input
TAiOUT pin function	A pulse output
Read from timer	The TAi register indicates undefined value
Write to timer	• While the timer counter is stopped or before the initial count source is input after starting to count, the value written to the TAi register is written to both reload register and the counter • While the timer counter is running, the value written to the TAi register is written to the reload register (it is transferred to the counter at the next reload timing)

Timer Ai Mode Register (i = 0 to 4) (pulse-width modulation mode)

b7	b6	b5	b4	b3	b2	b1	b0
					0	1	1

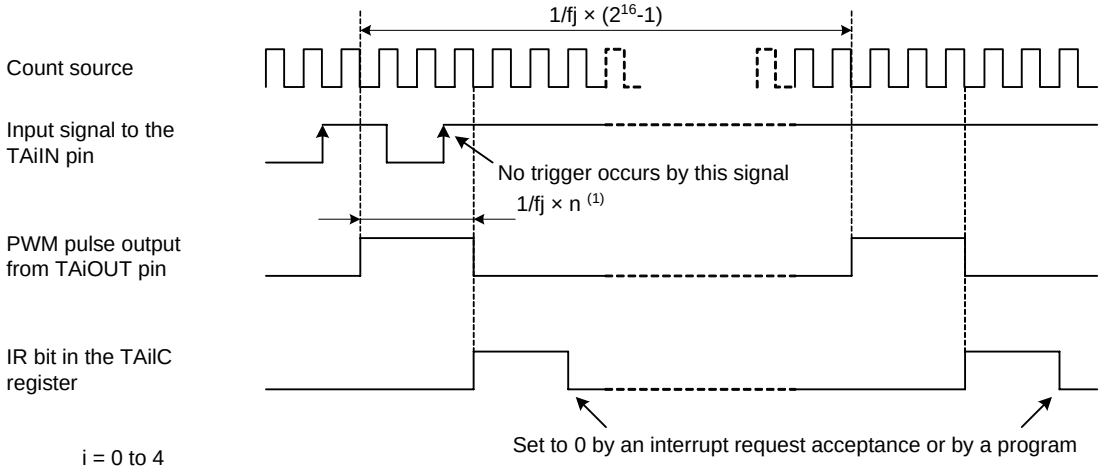
Symbol	Address	Reset Value
TA0MR to TA4MR	0356h, 0357h, 0358h, 0359h, 035Ah	0000 0000b

Bit Symbol	Bit Name	Function	RW
TMOD0	Operating Mode Select Bit	b1 b0 1 1 : Pulse-width modulation (PWM) mode	RW
TMOD1			RW
— (b2)	Reserved	Should be written with 0	RW
MR1	External Trigger Select Bit (1)	0: Falling edge of input signal to the TAiIN pin 1: Rising edge of input signal to the TAiIN pin	RW
MR2	Trigger Select Bit	0: The TAiS bit enabled 1: Selected using bits TAiTGH and TAiTGL	RW
MR3	16-/8-bit PWM Mode Select Bit	0: 16-bit pulse-width modulator 1: 8-bit pulse-width modulator	RW
TCK0	Count Source Select Bit	b7 b6 0 0 : f1 0 1 : f8 1 0 : f2n 1 1 : fC32	RW
TCK1			RW

Note:
1. The MR1 bit setting is enabled only when bits TAiTGH and TAiTGL in the TRGSR register are set to 00b (input to the TAiIN pin). This bit can be set to either 0 or 1 when bits TAiTGH and TAiTGL are set to 01b (the overflow or underflow of TB2), 10b (the overflow or underflow of TAi), or 11b (the overflow or underflow of TAi).

Figure 15.16 Registers TA0MR to TA4MR in Pulse-width Modulation Mode

When the reload register is set to 0003h and an external trigger (the rising edge of an input signal applied to the TAiIN pin) is selected

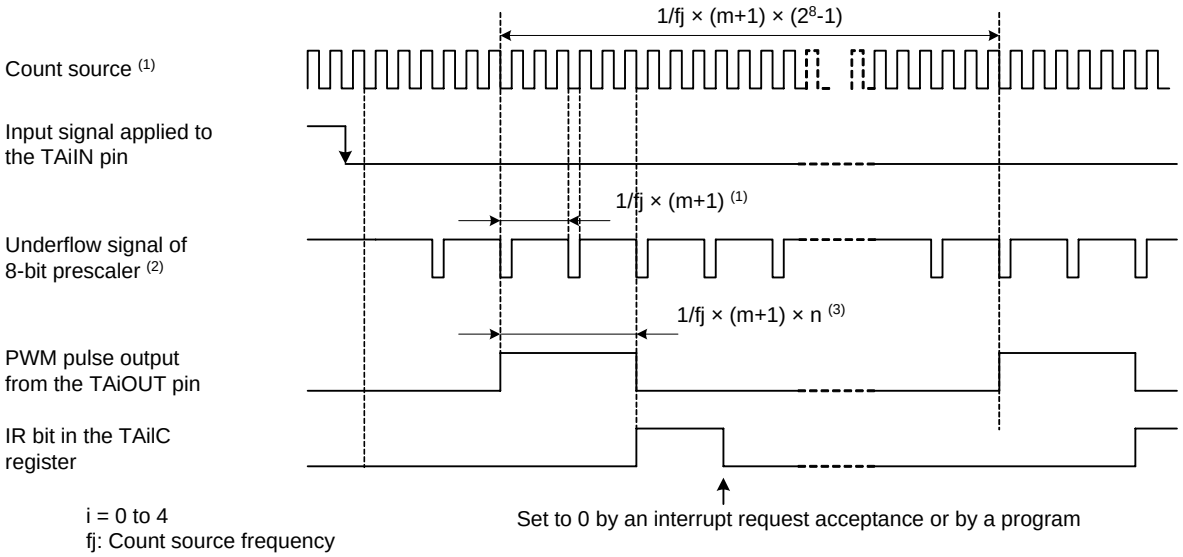

1/fj × (2^16-1)
Count source
Input signal to the TAiIN pin
No trigger occurs by this signal
1/fj × n (1)
PWM pulse output from TAiOUT pin
IR bit in the TAiIC register
Set to 0 by an interrupt request acceptance or by a program

i = 0 to 4
fj: Count source frequency
(f1, f8, f2n, fC32)

Note:
1. n = 0000h to FFFEh

Figure 15.17 16-bit Pulse-width Modulator Operation

When the upper byte of the reload register is set to 02h, the lower byte is set to 02h and an external trigger (the falling edge of an input signal applied to the TAiIN pin) is selected


1/fj × (m+1) × (2^8-1)
Count source (1)
Input signal applied to the TAiIN pin
1/fj × (m+1) (1)
Underflow signal of 8-bit prescaler (2)
1/fj × (m+1) × n (3)
PWM pulse output from the TAiOUT pin
IR bit in the TAiIC register
Set to 0 by an interrupt request acceptance or by a program

i = 0 to 4
fj: Count source frequency
(f1, f8, f2n, fC32)

Notes:
1. The 8-bit prescaler counts a count source.
2. The 8-bit pulse-width modulator counts underflow signals of the 8-bit prescaler.
3. m = 00h to FFh, n = 00h to FEh

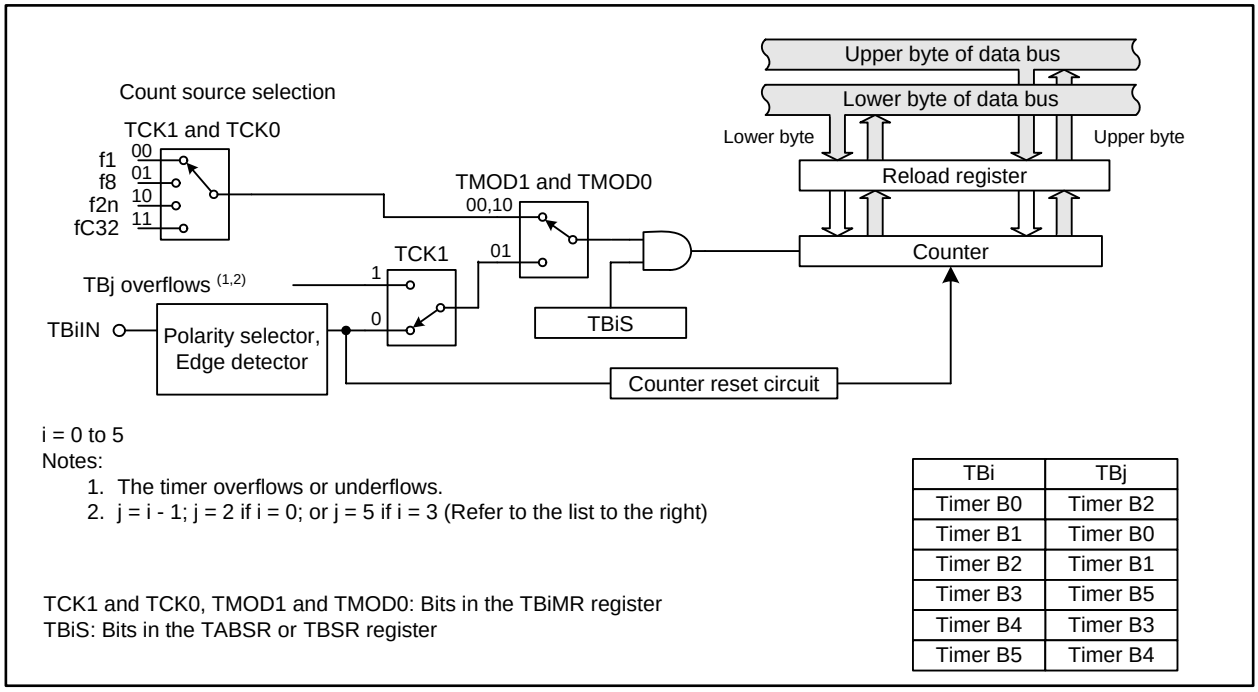
Figure 15.18 8-bit Pulse-width Modulator Operation

15.2 Timer B

Figure 15.19 shows a block diagram of the timer B and Figure 15.20 to Figure 15.23 show registers associated with the timer B.

The timer B supports three modes shown as below. A mode is selected using bits TMOD1 and TMOD0 in the TBiMR register (i = 0 to 5).

- Timer mode: The timer counts an internal count source
- Event counter mode: The timer counts an external pulse or an overflow and underflow of other timers
- Pulse period/pulse-width measure mode: The timer measures pulse period or pulse width of an external signal



i = 0 to 5

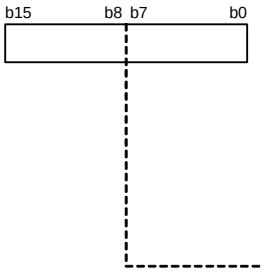
Notes:
1. The timer overflows or underflows.
2. j = i - 1; j = 2 if i = 0; or j = 5 if i = 3 (Refer to the list to the right)

TCK1 and TCK0, TMOD1 and TMOD0: Bits in the TBiMR register
TBiS: Bits in the TABSR or TBSR register

TBi	TBj
Timer B0	Timer B2
Timer B1	Timer B0
Timer B2	Timer B1
Timer B3	Timer B5
Timer B4	Timer B3
Timer B5	Timer B4

Figure 15.19 Timer B Block Diagram

Timer Bi Register (i = 0 to 5) (1)



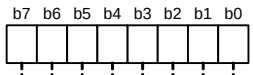
Symbol	Address	Reset Value
TB0 to TB2	0351h-0350h, 0353h-0352h, 0355h-0354h	Undefined
TB3 to TB5	0311h-0310h, 0313h-0312h, 0315h-0314h	Undefined

Mode	Function	Setting Range	RW
Timer Mode	Divides the count source by n+1 (n = setting value)	0000h to FFFFh	RW
Event Counter Mode	Divides the count source by n+1 (n = setting value) (2)	0000h to FFFFh	RW
Pulse Period/Pulse-width Measure Mode	Increments the count with the count source between one valid edge and another of TBiIN input pulse	—	RO

Notes:
1. A 16-bit read/write access to this register should be performed.
2. The TBi register counts an external input pulse or an overflow and underflow of other timers.

Figure 15.20 Registers TB0 to TB5

Timer Bi Mode Register (i = 0 to 5)



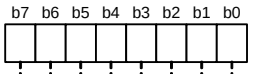
Symbol	Address	Reset Value
TB0MR to TB2MR	035Bh, 035Ch, 035Dh	00XX 0000b
TB3MR to TB5MR	031Bh, 031Ch, 031Dh	00XX 0000b

Bit Symbol	Bit Name	Function	RW
TMOD0	Operating Mode Select Bit	b1 b0 0 0 : Timer mode 0 1 : Event counter mode 1 0 : Pulse period measure mode, pulse-width measure mode 1 1 : Do not use this combination	RW
TMOD1			RW
MR0	—	Function varies according to the operating mode (1, 2)	RW
MR1			RW
MR2			RW
MR3			RW
TCK0	Count Source Select Bit	Function varies according to the operating mode	RW
TCK1			RW

Notes:
1. The MR2 bit is available for registers TB0MR and TB3MR only.
2. The MR2 bit in registers TB1MR, TB2MR, TB4MR and TB5MR are unavailable on this MCU. This bit should be written with 0 and read as undefined value.

Figure 15.21 Registers TB0MR to TB5MR

Count Start Register

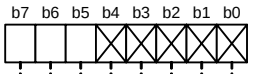


Symbol	Address	Reset Value
TABSR	0340h	0000 0000b

Bit Symbol	Bit Name	Function	RW
TA0S	Timer A0 Count Start Bit	0: Stop counter 1: Start counter	RW
TA1S	Timer A1 Count Start Bit	0: Stop counter 1: Start counter	RW
TA2S	Timer A2 Count Start Bit	0: Stop counter 1: Start counter	RW
TA3S	Timer A3 Count Start Bit	0: Stop counter 1: Start counter	RW
TA4S	Timer A4 Count Start Bit	0: Stop counter 1: Start counter	RW
TB0S	Timer B0 Count Start Bit	0: Stop counter 1: Start counter	RW
TB1S	Timer B1 Count Start Bit	0: Stop counter 1: Start counter	RW
TB2S	Timer B2 Count Start Bit	0: Stop counter 1: Start counter	RW

Figure 15.22 TABSR Register

Count Start Register for Timers B3, B4 and B5



Symbol	Address	Reset Value
TBSR	0300h	000X XXXXb

Bit Symbol	Bit Name	Function	RW
— (b4-b0)	No register bits; should be written with 0 and read as undefined value		—
TB3S	Timer B3 Count Start Bit	0: Stop counting 1: Start counting	RW
TB4S	Timer B4 Count Start Bit	0: Stop counting 1: Start counting	RW
TB5S	Timer B5 Count Start Bit	0: Stop counting 1: Start counting	RW

Figure 15.23 TBSR Register

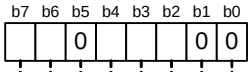
15.2.1 Timer Mode

In timer mode, the timer counts an internally generated count source. Table 15.6 lists specifications of timer mode. Figure 15.24 shows registers TB0MR to TB5MR in this mode.

Table 15.6 Timer Mode Specifications (i = 0 to 5)

Item	Specification
Count sources	f1, f8, f2n, or fC32
Count operations	• Decrement counting • If the timer counter underflows, the reload register setting is reloaded into the counter to resume counting
Divide ratio	$\frac{1}{n+1}$ n: TBi register setting value, 0000h to FFFFh
Count start condition	The TBiS bit in the TABSR or TBSR register is set to 1 (count starts)
Count stop condition	The TBiS bit in the TABSR or TBSR register is set to 0 (count stops)
Interrupt request generating timing	When the timer counter underflows
TBiIN pin function	A programmable I/O port
Read from timer	The TBi register indicates a counter value
Write to timer	• While the timer counter is stopped or before the initial count source is input after starting to count, the value written to the TBi register is written to both reload register and the counter • While the timer counter is running, the value written to the TBi register is written to the reload register (It is transferred to the counter at the next reload timing)

Timer Bi Mode Register (i = 0 to 5) (timer mode)



Symbol	Address	Reset Value
TB0MR to TB2MR	035Bh, 035Ch, 035Dh	00XX 0000b
TB3MR to TB5MR	031Bh, 031Ch, 031Dh	00XX 0000b

Bit Symbol	Bit Name	Function	RW
TMOD0	Operating Mode Select Bit	b1 b0 0 0 : Timer mode	RW
TMOD1			RW
MR0	Disabled in timer mode. Can be set to 0 or 1		RW
MR1			RW
MR2	In registers TB0MR and TB3MR; Reserved; should be written with a 0		RW
	In registers TB1MR, TB2MR, TB4MR, and TB5MR; No register bit; should be written with 0 and read as undefined value		—
MR3	Disabled in timer mode. Should be written with a 0 and read as undefined value		—
TCK0	Count Source Select Bit	b7 b6 0 0 : f1 0 1 : f8 1 0 : f2n 1 1 : fC32	RW
TCK1			RW

Figure 15.24 Registers TB0MR to TB5MR in Timer Mode

15.2.2 Event Counter Mode

In event counter mode, the timer counts an external signal or an overflow and underflow of other timers. Table 15.7 lists specifications of event counter mode. Figure 15.25 shows the TBiMR register (i = 0 to 5) in this mode.

Table 15.7 Event Counter Mode Specifications (i = 0 to 5)

Item	Specification
Count sources	• External signal applied to the TBiIN pin (valid edge is selectable among the falling edge, the rising edge or the both by a program) • The overflow or underflow signal of TBj (j = i - 1; j = 2 if i = 0; or j = 5 if i = 3)
Count operations	• Decrement counting • If the timer counter underflows, the reload register setting is reloaded into the counter to resume counting
Divide ratio	$\frac{1}{n+1}$ n: TBi register setting value, 0000h to FFFFh
Count start condition	The TBiS bit in the TABSR or TBSR register is set to 1 (count starts)
Count stop condition	The TBiS bit in the TABSR or TBSR register is set to 0 (count stops)
Interrupt request generation timing	When the timer counter underflows
TBiIN pin function	A programmable I/O port or a count source input
Read from timer	The TBi register indicates a counter value
Write to timer	• While the timer counter is stopped or before the initial count source is input after starting to count, the value written to the TBi register is written to both reload register and the counter • While the timer counter is running, the value written to the TBi register is written to the reload register (it is transferred to the counter at the next reload timing)

Timer Bi Mode Register (i = 0 to 5) (event counter mode)

b7	b6	b5	b4	b3	b2	b1	b0
		0				0	1

Symbol	Address	Reset Value
TB0MR to TB2MR	035Bh, 035Ch, 035Dh	00XX 0000b
TB3MR to TB5MR	031Bh, 031Ch, 031Dh	00XX 0000b

Bit Symbol	Bit Name	Function	RW
TMOD0	Operating Mode Select Bit	b1 b0 0 1 : Event counter mode	RW
TMOD1			RW
MR0	Count Polarity Select Bit (1)	b3 b2 0 0 : Count falling edges 0 1 : Count rising edges 1 0 : Count both edges 1 1 : Do not use this combination	RW
MR1			RW
MR2	In registers TB0MR and TB3MR; Reserved; should be written with 0		RW
	In registers TB1MR, TB2MR, TB4MR, and TB5MR; No register bit; should be written with 0 and read as undefined value		—
MR3	Disabled in event counter mode. Should be written with 0 and read as undefined value		—
TCK0	Disabled in event counter mode. Can be set to 0 or 1		RW
TCK1	Event Clock Select Bit	0: Input signal to the TBiIN pin 1: The overflow or underflow of TBj (2)	RW

Notes:
1. These bit settings are enabled when the TCK1 bit is set to 0.
   When the TCK1 is set to 1, these bits can be set to either 0 or 1.
2. j = i - 1; j = 2 if i = 0; or j = 5 if i = 3.

Figure 15.25 Registers TB0MR to TB5MR in Event Counter Mode

15.2.3 Pulse Period/Pulse-width Measure Mode

In pulse period/pulse-width measure mode, the timer measures pulse period or pulse width of an external signal. Table 15.8 lists specifications of pulse period/pulse-width measure mode. Figure 15.26 shows registers TB0MR to TB5MR in this mode. Figure 15.27 and Figure 15.28 respectively show an operation example of pulse period measurement and pulse-width measurement.

Table 15.8 Pulse Period/Pulse-width Measure Mode Specifications (i = 0 to 5)

Item	Specification
Count sources	f1, f8, f2n, or fC32
Count operations	• Increment counting • The counter value is transferred to the reload register on the valid edge of a pulse to be measured, then it is set to 0000h to resume counting
Count start condition	The TBiS bit in the TABSR or TBSR register is set to 1 (count starts)
Count stop condition	The TBiS bit in the TABSR or TBSR register is set to 0 (count stops)
Interrupt request generating timing	• On the valid edge of a pulse to be measured (1) • When the timer counter overflows (when the MR3 bit in the TBiMR register becomes 1 (overflow). (2))
TBiIN pin function	A pulse input to be measured
Read from timer	The TBi register indicates a reload register value (measurement results) (3)
Write to timer	The value written to the TBi register is written to neither the reload register nor the counter

Notes:
1. No interrupt request is generated when the pulse to be measured is applied on the initial valid edge after the timer counter starts.
2. To set the MR3 bit to 0 (no overflow), wait one or more count source cycles to write to the TBiMR register after the MR3 bit becomes 1 (overflow), while the TBiS bit is set to 1 (count starts).
3. The TBi register indicates undefined value until the pulse to be measured is applied on the second valid edge after the timer counter starts.

Timer Bi Mode Register (i = 0 to 5) (pulse period/pulse-width measure mode)

Symbol	Address	Reset Value
TB0MR to TB2MR	035Bh, 035Ch, 035Dh	00XX 0000b
TB3MR to TB5MR	031Bh, 031Ch, 031Dh	00XX 0000b

b7	b6	b5	b4	b3	b2	b1	b0
						1	0

Bit Symbol	Bit Name	Function	RW
TMOD0	Operating Mode Select Bit	b1 b0 1 0 : Pulse period/pulse-width measure mode	RW
TMOD1			RW
MR0	Measure Mode Select Bit (1)	b3 b2 0 0 : Pulse period measurement 1 0 1 : Pulse period measurement 2 1 0 : Pulse-width measurement 1 1 : Do not use this combination	RW
MR1			RW
MR2	In registers TB0MR and TB3MR; Reserved; should be written with 0		RW
	In registers TB1MR, TB2MR, TB4MR, and TB5MR; No register bit; should be written with 0 and read as undefined value		—
MR3	Timer Bi Overflow Flag (2)	0: No overflow 1: Overflow	RO
TCK0	Count Source Select Bit	b7 b6 0 0 : f1 0 1 : f8 1 0 : f2n 1 1 : fC32	RW
TCK1			RW

Notes:

1. Bits MR1 and MR0 select from the following measure modes:
   Pulse period measurement 1 (bits MR1 and MR0 = 00b):
   Measures between a falling edge and the next falling edge of a pulse
   Pulse period measurement 2 (bits MR1 and MR0 = 01b):
   Measures between a rising edge and the next rising edge of a pulse
   Pulse-width measurement (bits MR1 and MR0 = 10b):
   Measures between a falling edge and the next rising edge of a pulse and between the rising edge and the next falling edge of the pulse
2. The MR3 bit is undefined when the timer is reset.
   To set the MR3 bit to 0 (no overflow), wait one or more count source cycles to write to the TBiMR register after the MR3 bit becomes 1 (overflow), while the TBiS bit in the TABSR or TBSR register is set to 1 (count starts).
   The MR3 bit cannot be set to 1 by a program.

Figure 15.26 Registers TB0MR to TB5MR in Pulse Period/Pulse-width Measure Mode

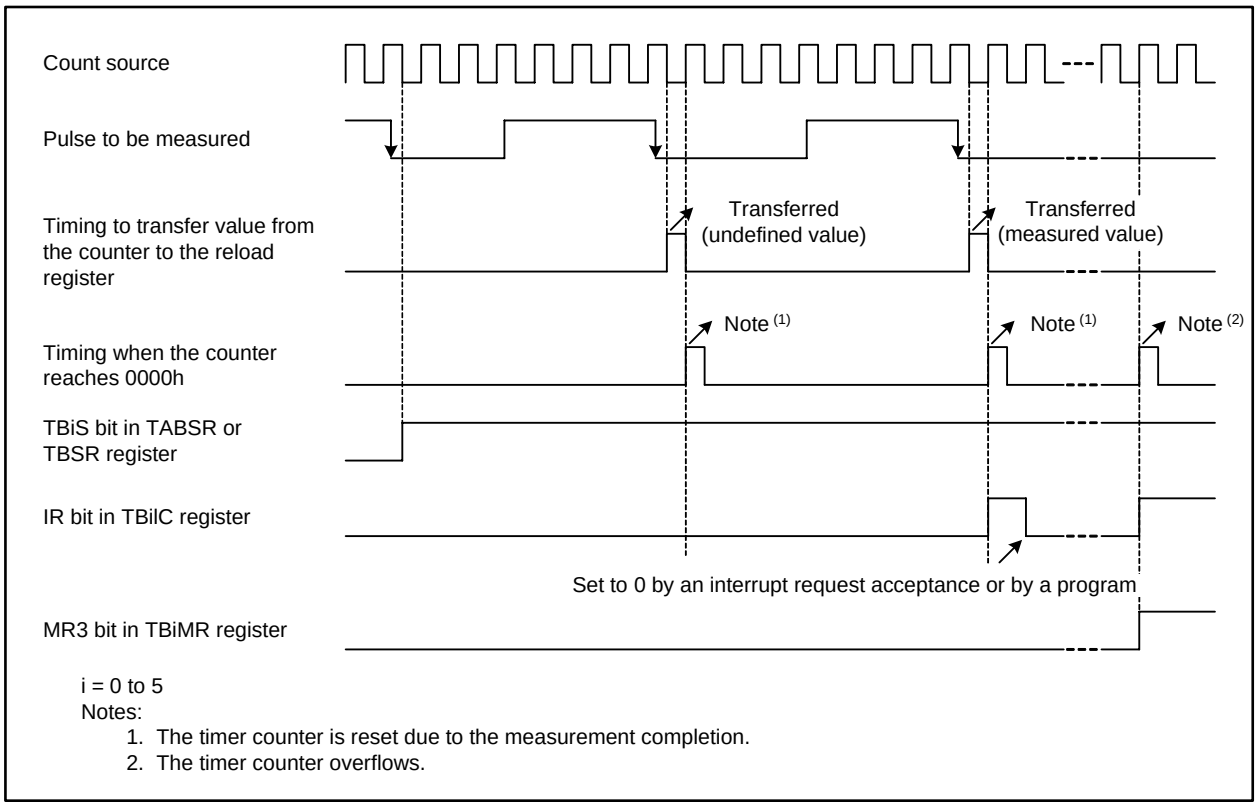

Count source
Pulse to be measured
Timing to transfer value from the counter to the reload register
Transferred (undefined value)
Transferred (measured value)
Timing when the counter reaches 0000h
Note (1)
Note (1)
Note (2)
TBiS bit in TABSR or TBSR register
IR bit in TBilC register
Set to 0 by an interrupt request acceptance or by a program
MR3 bit in TBiMR register
i = 0 to 5
Notes:
1. The timer counter is reset due to the measurement completion.
2. The timer counter overflows.

Figure 15.27 Operation Example in Pulse Period Measurement

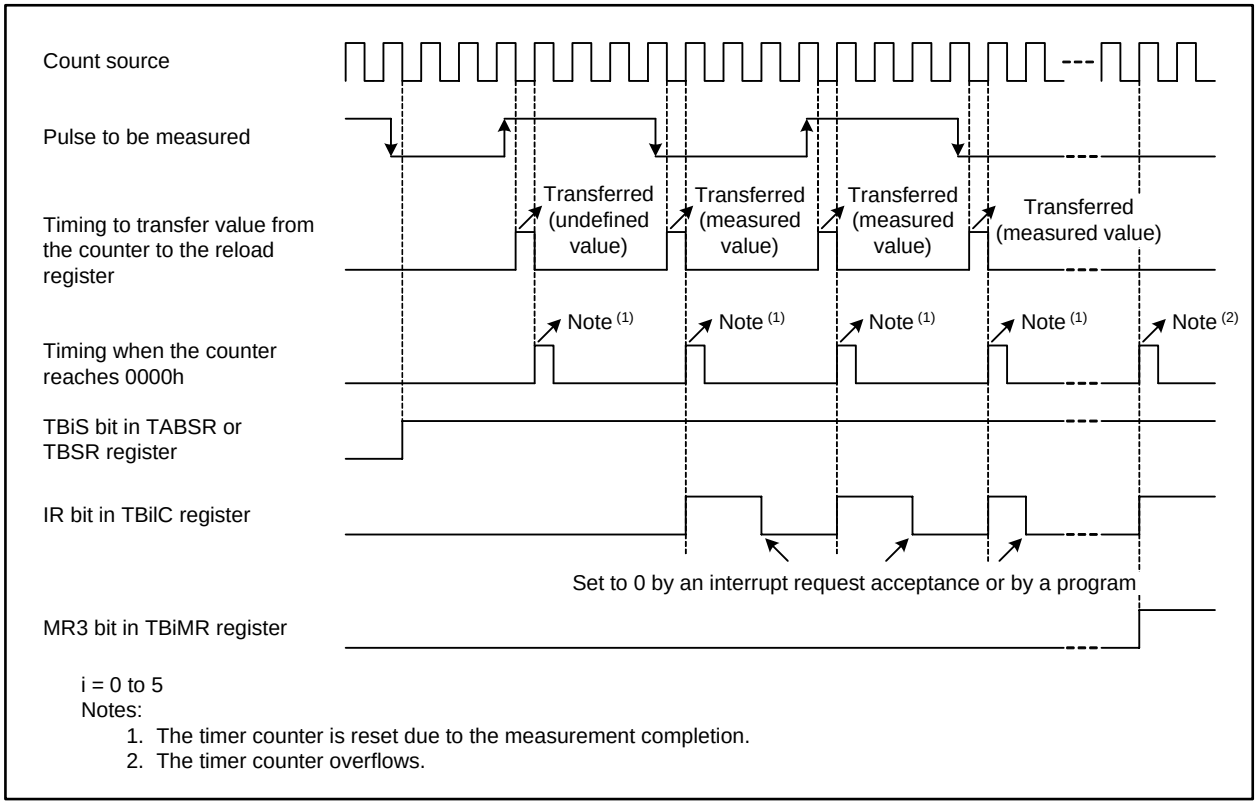

Count source
Pulse to be measured
Timing to transfer value from the counter to the reload register
Transferred (undefined value)
Transferred (measured value)
Transferred (measured value)
Transferred (measured value)
Timing when the counter reaches 0000h
Note (1)
Note (1)
Note (1)
Note (1)
Note (2)
TBiS bit in TABSR or TBSR register
IR bit in TBilC register
Set to 0 by an interrupt request acceptance or by a program
MR3 bit in TBiMR register
i = 0 to 5
Notes:
1. The timer counter is reset due to the measurement completion.
2. The timer counter overflows.

Figure 15.28 Operation Example in Pulse-width Measurement

15.3 Notes on Timers

15.3.1 Timer A and Timer B

All timers are stopped after a reset. To restart timers, configure parameters such as operating mode, count source, and counter value, then set the TAiS bit (i = 0 to 4) or TBjS bit (j = 0 to 5) in the TABSR or TBSR register to 1 (count starts).

The following registers and bits should be set while the TAiS bit or TBjS bit is 0 (count stops):

- Registers TAiMR and TBjMR
- The UDF register
- Bits TAZIE, TA0TGL, and TA0TGH in the ONSF register
- The TRGSR register

15.3.2 Timer A

15.3.2.1 Timer Mode

- While the timer counter is running, the TAi register indicates a counter value at any given time. However, FFFFh is read while reloading is in progress. A set value is read if the TAi register is set while the timer counter is stopped.

15.3.2.2 Event Counter Mode

- While the timer counter is running, the TAi register indicates a counter value at any given time. However, FFFFh is read if the timer counter underflows or 0000h if overflows while reloading is in progress. A set value is read if the TAi register is set while the timer counter is stopped.

15.3.2.3 One-shot Timer Mode

- If the TAiS bit in the TABSR register is set to 0 (count stops) while the timer counter is running, the following operations are performed:
  - The timer counter stops and the setting value of the TAi register is reloaded.
  - A low signal is output at the TAiOUT pin.
  - The IR bit in the TAiIC register becomes 1 (interrupts requested) after one CPU clock cycle.

- One-shot timer is operated by an internal count source. When the trigger is an input to the TAiIN pin, the signal is output with a maximum of one count source clock delay after a trigger input to the TAiIN pin.

- The IR bit becomes 1 by any of the settings below. To use the timer Ai interrupt, set the IR bit to 0 after one of the settings below is done:
  - Select one-shot timer mode after a reset.
  - Switch the operating mode from timer mode to one-shot timer mode.
  - Switch the operating mode from event counter mode to one-shot timer mode.

- If a retrigger occurs while counting, the timer counter decrements by one, reloads the setting value of the TAi register, and then continues counting. To generate a retrigger while counting, wait one or more count source cycles after the last trigger is generated.

- When an external trigger input is selected to start counting in timer A one-shot mode, do not provide an external retrigger for 300 ns before the timer counter reaches 0000h. Otherwise, it may stop counting.

15.3.2.4 Pulse-width Modulation Mode

- The IR bit becomes 1 by any of the settings below. To use the timer Ai interrupt (i = 0 to 4), set the IR bit to 0 after one of the settings below is done:
  - Select pulse-width modulation mode after a reset.
  - Switch the operating mode from timer mode to pulse-width modulation mode.
  - Switch the operating mode from event counter mode to pulse-width modulation mode.

- If the TAiS bit in the TABSR register is set to 0 (count stops) while PWM pulse is output, the following operations are performed:
  - The timer counter stops.
  - The output level at the TAiOUT pin changes from high to low. The IR bit becomes 1.
  - When a low signal is output at the TAiOUT pin, it remains unchanged. The IR bit does not change, either.

15.3.3 Timer B

15.3.3.1 Timer Mode and Event Counter Mode

- While the timer counter is running, the TBj register (j = 0 to 5) indicates a counter value at any given time. However, FFFFh is read while reloading is in progress. A set value is read if the TBj register is set while the timer counter is stopped.

15.3.3.2 Pulse Period/Pulse-width Measure Mode

- To set the MR3 bit in the TBjMR register to 0 (no overflow), wait one or more count source cycles to write to the TBjMR register after the MR3 bit becomes 1 (overflow), while the TBjS bit is set to 1 (count starts).

- Use the IR bit in the TBjIC register to detect overflow. The MR3 bit is used only to determine an interrupt request source within the interrupt handler.

- The counter value is undefined when the timer counter starts. Therefore, the timer counter may overflow before a pulse to be measured is applied on the initial valid edge and cause a timer Bj interrupt request to be generated.

- When the pulse to be measured is applied on the initial valid edge after the timer counter starts, an undefined value is transferred to the reload register. At this time, the timer Bj interrupt request is not generated.

- The IR bit may become 1 (interrupt requested) by changing bits MR1 and MR0 in the TBjMR register after the timer counter starts. However, if the same value is rewritten to bits MR1 and MR0, the IR bit is not changed.

- Pulse width is repeatedly measured in pulse-width measure mode. Whether the measurement result is high-level width or not is determined by a program.

- If an overflow occurs simultaneously when a pulse is applied on the valid edge, this pulse is not recognized since an interrupt request is generated only once. Do not let an overflow occur in pulse period measure mode.

- In pulse-width measure mode, determine whether an interrupt source is a pulse applied on the valid edge or an overflow by reading the port level in the TBj interrupt handler.

16. Three-phase Motor Control Timers

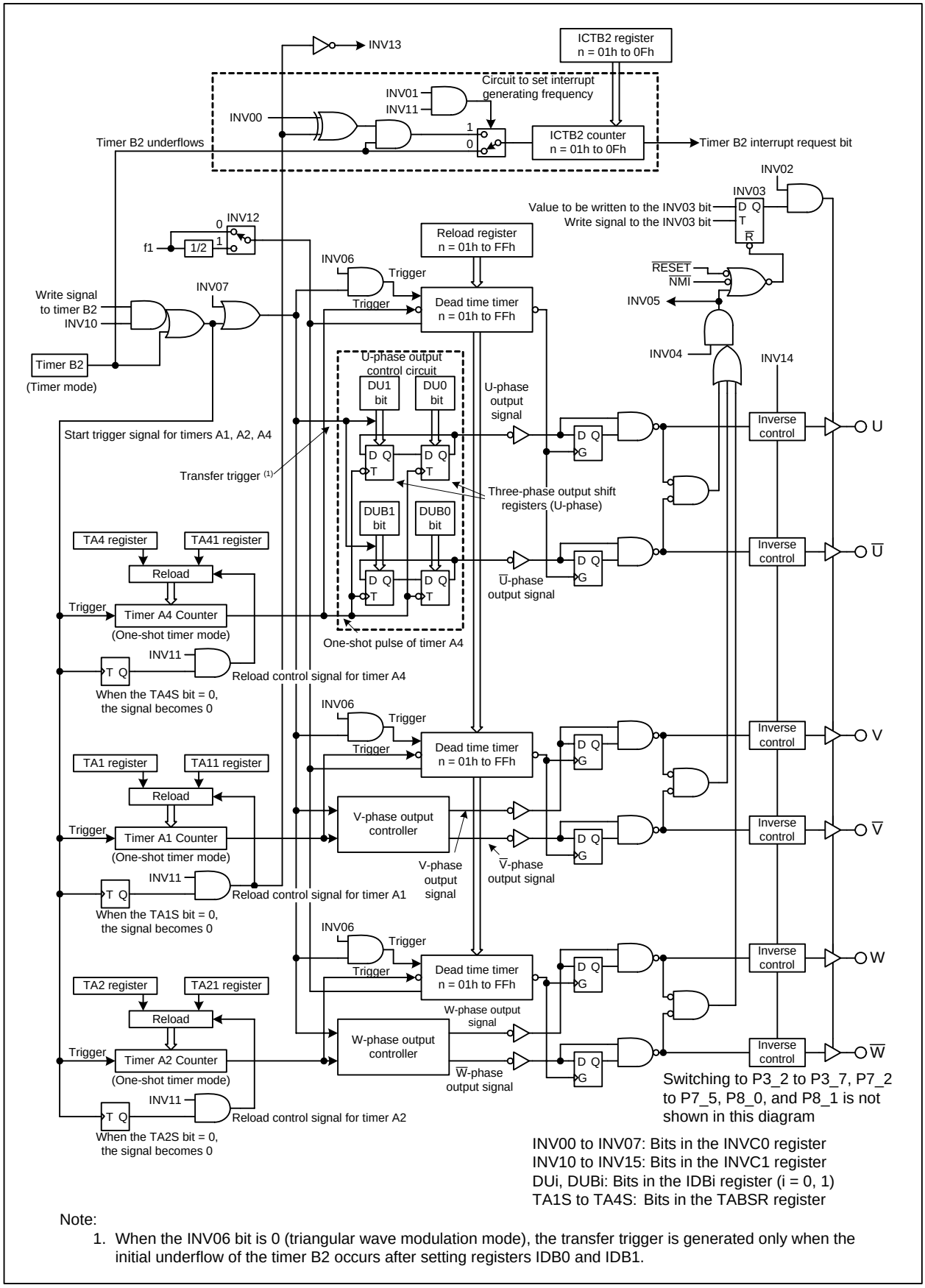
A three-phase motor driving waveform can be output by using timers A1, A2, A4, and B2. The three-phase motor control timers are enabled by setting the INV02 bit in the INVC0 register to 1. Timer B2 is used for carrier wave control, and timers A1, A2, and A4 for three-phase PWM output (U, $\overline{U}$, V, $\overline{V}$, W, and $\overline{W}$) control. Table 16.1 lists specifications of the three-phase motor control timers and Figure 16.1 shows its block diagram. Figure 16.2 to Figure 16.6 show registers associated with this function.

Table 16.1 Three-phase Motor Control Timers Specifications

Item	Specification
Three-phase PWM waveform output pins	Six pins: U, $\overline{U}$, V, $\overline{V}$, W, and $\overline{W}$
Forced cutoff (1)	A low input to the $\overline{NMI}$ pin
Timers to be used	Timers A4, A1, and A2 (used in one-shot timer mode): Timer A4: U- and $\overline{U}$-phases waveform control Timer A1: V- and $\overline{V}$-phases waveform control Timer A2: W- and $\overline{W}$-phases waveform control Timer B2 (used in timer mode) Carrier wave cycle control Dead time timer (three 8-bit timers share a reload register): Dead time control
Output waveform	Triangular wave modulation and sawtooth wave modulation • Output of a high or a low waveform for one cycle • Separately settable levels of high side and low side
Carrier wave cycle	Triangular wave modulation: count source × (m+1) × 2 Sawtooth wave modulation: count source × (m+1) m: TB2 register setting value, 0000h to FFFFh Count source: f1, f8, f2n, or fC32
Three-phase PWM output width	Triangular wave modulation: count source × n × 2 Sawtooth wave modulation: count source × n n: Setting value of registers TA4, TA1, and TA2 (registers TA4, TA41, TA1, TA11, TA2, and TA21 when the INV11 bit in the INVC1 register is set to 1), 0001h to FFFFh Count source: f1, f8, f2n, or fC32
Dead time (width)	Count source × p or no dead time p: DTT register setting value, 01h to FFh Count source: f1 or f1 divided by 2
Active level	Selectable either active high or active low
Simultaneous conduction prevention	Function to detect simultaneous turn-on signal outputs, function to disable signal output when simultaneous turn-on signal outputs are detected
Interrupt frequency	Selectable from one through 15 time- carrier wave cycle-to-cycle basis for the timer B2 interrupt

Note:

1. Forced cutoff by the signal input to the $\overline{NMI}$ pin is available when the PM24 bit in the PM2 register is set to 1 (NMI enabled), the INV02 bit in the INVC0 register is set to 1 (the three-phase motor control timers used), and the INV03 bit is set to 1 (the three-phase motor control timer output enabled).



Note:

1. When the INV06 bit is 0 (triangular wave modulation mode), the transfer trigger is generated only when the initial underflow of the timer B2 occurs after setting registers IDB0 and IDB1.

Figure 16.1 Three-phase Motor Control Timers Block Diagram

Three-phase PWM Control Register 0 (1)

b7	b6	b5	b4	b3	b2	b1	b0

Symbol	Address	Reset Value
INVC0	0308h	0000 0000b

Bit Symbol	Bit Name	Function	RW
INV00	ICTB2 Count Condition Select Bit (2)	b1 b0 0 X : The underflow of timer B2 1 0 : The underflow of timer B2 when the reload control signal for timer A1 is set to 0 (3) 1 1 : The underflow of timer B2 when the reload control signal for timer A1 is set to 1 (3, 4)	RW
INV01			RW
INV02	Three-phase Motor Control Timers Select Bit	0: Do not use this function 1: Use this function (5, 6, 7)	RW
INV03	Three-phase Motor Control Timer Output Control Bit	0: Disables the three-phase motor control timer output (7) 1: Enables the three-phase motor control timer output (8)	RW
INV04	Simultaneous Conduction Prevention Bit	0: Ignores simultaneous turn-on signal output 1: Disables simultaneous turn-on signal output	RW
INV05	Simultaneous Conduction Detection Flag	0: Not detected 1: Detected (9)	RO
INV06	Modulation Mode Select Bit	0: Triangular wave modulation mode 1: Sawtooth wave modulation mode (10)	RW
INV07	Software Trigger Select Bit	A transfer trigger is generated when this bit is set to 1. When the INV06 bit is set to 1, another trigger to the dead time timer is also generated. This bit is read as 0	RW

Notes:
1. This register should be set after the PRC1 bit in the PRCR register is set to 1 (write enabled). Bits INV00 to INV02 and INV06 should be rewritten while timers A1, A2, A4, and B2 are stopped.
2. This bit is enabled when the INV11 bit in the INVC1 register is set to 1 (three-phase mode 1). When the INV11 bit is set to 0 (three-phase mode 0), the ICTB2 counter increments by one every time the timer B2 underflows irrespective of the INV00 and INV01 bit settings.
3. Set the ICTB2 register before setting the INV01 bit to 1. The timer A1 count start flag should be set to 1 before the initial timer B2 underflow occurs.
4. When the INV00 bit is set to 1, the first interrupt is generated if the timer B2 underflows n-1 times. (n is the value set in the ICTB2 counter). Subsequent interrupts are generated every n times the timer B2 underflows.
5. The INV02 bit should be set to 1 to operate the dead time timer, U-, V-, and W-phase output control circuits, and the ICTB2 counter.
6. After setting the INV02 bit to 1, pins should be configured first by the IOBC register then by the output function select register.
7. When the INV02 bit is set to 1 and the INV03 bit to 0, pins U, $\overline{U}$, V, $\overline{V}$, W, and $\overline{W}$, including shared pins set by another output function, become high-impedance.
8. The INV03 bit becomes 0 when any of the following occurs:
   -Reset
   -Signals of both sides high and low are simultaneously switched active when the INV04 bit is set to 1
   -The INV03 bit is set to 0 by a program
   -The $\overline{NMI}$ pin goes from high to low when the PM24 bit in the PM2 register is set to 1 (NMI enabled)
9. This bit cannot be set to 1 by a program. The INV04 bit should be set to 0 to set this bit to 0.
10. When the INV06 bit is set to 1, the INV11 bit in the INVC1 register should be set to 0 (three-phase mode 0). In this case, the PWCON bit in the TB2SC register should be set to 0 (timer B2 register reloaded if the timer B2 underflows).

Figure 16.2 INVC0 Register

Three-phase PWM Control Register 1 (1)

b7	b6	b5	b4	b3	b2	b1	b0
0							

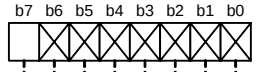
Symbol: INVC1 | Address: 0309h | Reset Value: 0000 0000b

Bit Symbol	Bit Name	Function	RW
INV10	Timers A1, A2, and A4 Start Trigger Select Bit	0: The underflow of timer B2 1: The underflow of timer B2 and a write to the TB2 register	RW
INV11	Timers A1-1, A2-1, and A4-1 Control Bit	0: Three-phase mode 0 (2, 3) 1: Three-phase mode 1	RW
INV12	Dead Time Timer Count Source Select Bit	0: f1 1: f1 divided-by-2	RW
INV13	Carrier Wave Detection Flag (4)	0: Timer A1 reload control signal is 0 1: Timer A1 reload control signal is 1	RO
INV14	Active Level Control Bit	0: Active low output 1: Active high output	RW
INV15	Dead Time Disable Bit	0: Enables dead time 1: Disables dead time	RW
INV16	Dead Time Timer Trigger Select Bit	0: Falling edge of a one-shot pulse of timer (A4, A1, and A2) (5) 1: Rising edge of the three-phase output shift register (phases U, V, and W)	RW
— (b7)	Reserved	Should be written with 0	RW

Notes:
1. This register should be set after the PRC1 bit in the PRCR register is set to 1 (write enabled). Bits INV00 to INV02 and INV06 should be rewritten while timers A1, A2, A4, and B2 are stopped.
2. When the INV06 bit in the INVC0 register is set to 1 (sawtooth wave modulation mode), the INV11 bit should be set to 0 (three-phase mode 0).
3. When the INV11 bit is set to 0, the PWCON bit in the TB2SC register should be set to 0 (timer B2 register reloaded if the timer B2 underflows).
4. This bit setting is enabled when the INV06 bit is set to 0 (triangular wave modulation mode) and the INV11 bit is set to 1 (three-phase mode 1).
5. If the following conditions are all met, the INV16 bit should be set to 1:
   -The INV15 bit is set to 0 (dead time enabled)
   -The Dij bit (i = U, V, or W; j = 0 to 1) has a different value from the DiBj bit whenever the INV03 bit is set to 1 (three-phase motor control timer output enabled); the high- and low-side output signals have always inverse level on the period other than dead time.
   If any of the conditions above are not met, the INV16 bit should be set to 0.

Figure 16.3 INVC1 Register

Three-phase Output Buffer Control Register (1)



Symbol	Address	Reset Value
IOBC	40097h	0XXX XXXXb

Bit Symbol	Bit Name	Function	RW
— (b6-b0)	No register bits; should be written with 0 and read as undefined value		—
TBSOUT	Three-phase Output Pin Select Bit (2)	0: Use pins U, $\overline{U}$, V, $\overline{V}$, W, and $\overline{W}$ of ports P7 and P8 1: Use pins U, $\overline{U}$, V, $\overline{V}$, W, and $\overline{W}$ of port P3	RW

Notes:
1. This register should be set after the PRC1 bit in the PRCR register is set to 1 (write enabled).
2. This bit should be set after the INV02 bit in the INVC0 register is set to 1. Then, the output function select register of corresponding port should be set. When the INV03 bit in the INVC0 register is set to 0, output pins for the three-phase motor control timers become high-impedance by the output enable control of output buffers. However, the output enable cannot be controlled only by the output function select register when more than two ports are assigned for output. Thus, a three-state output buffer should be selected using the TBSOUT bit.

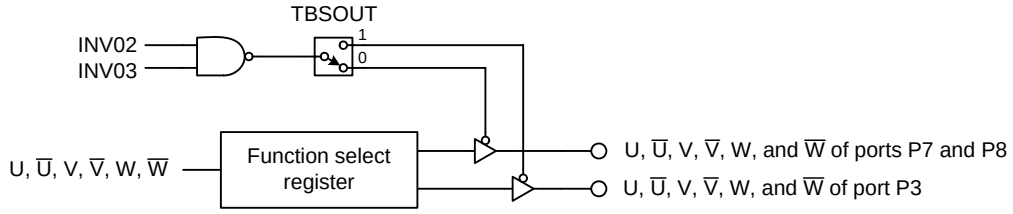
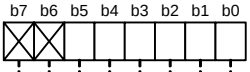

TBSOUT
INV02
INV03
1
0
U, $\overline{U}$, V, $\overline{V}$, W, $\overline{W}$
Function select register
U, $\overline{U}$, V, $\overline{V}$, W, and $\overline{W}$ of ports P7 and P8
U, $\overline{U}$, V, $\overline{V}$, W, and $\overline{W}$ of port P3

Figure 16.4 IOBC Register

Three-phase Output Buffer Register i (i = 0, 1) (1)



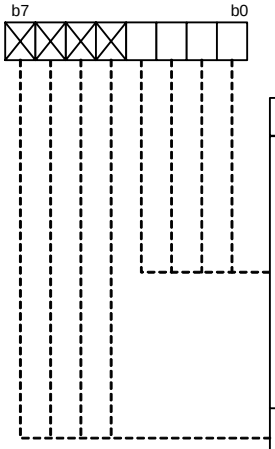
Symbol	Address	Reset Value
IDB0, IDB1	030Ah, 030Bh	XX11 1111b

Bit Symbol	Bit Name	Function	RW
DUi	U-phase Output Buffer i	These bits should be written with an output level of the three-phase output shift register. The written value is reflected in each turn-on signal as follows: 0: Active (ON) 1: Inactive (OFF) The bits are read as the value of the three-phase output shift register	RW
DUBi	$\overline{U}$-phase Output Buffer i		RW
DVi	V-phase Output Buffer i		RW
DVBi	$\overline{V}$-phase Output Buffer i		RW
DWi	W-phase Output Buffer i		RW
DWBi	$\overline{W}$-phase Output Buffer i		RW
— (b7-b6)	No register bits; should be written with 0 and read as undefined value		—

Note:
1. Values of registers IDB0 and IDB1 are transferred to the three-phase output shift register by a transfer trigger. The initial output signal level of each phase is determined by the value written in the IDB0 register after the transfer trigger occurs. After that, the output signal level is determined by the value written in the IDB1 register on the falling edge of a one-shot pulse of timers A1, A2, and A4.

Figure 16.5 Registers IDB0 and IDB1

Timer B2 Interrupt Generating Frequency Set Counter (1, 2, 3)



Symbol	Address	Reset Value
ICTB2	030Dh	Undefined

Function	Setting Range	RW
-When the INV01 bit = 0 (the ICTB2 counter increments every time the timer B2 underflows), the timer B2 interrupt request is generated every nth times the timer B2 underflows -When the INV01 bit = 1 (the ICTB2 counter increments when the timer A1 reload control signal is set to 0 or 1 and the timer B2 underflows), the timer B2 interrupt request is generated every nth times the timer B2 underflows when the timer A1 reload control signal is 0 or 1 (n = setting value)	01h to 0Fh	WO
No register bits; should be written with 0		—

Notes:
1. The MOV instruction should be used to set the ICTB2 register.
2. When the INV01 bit in the INVC0 register is set to 1, set this register when the TB2S bit in the TABSR register is 0 (timer B2 count stops). When the INV01 bit is set to 0, this register can be set irrespective of the TB2S bit setting. However, do not set the register when the timer B2 underflows.
3. When the INV00 bit in the INVC0 register is set to 1, the first interrupt is generated if the timer B2 underflows n-1 times. Subsequent interrupts are generated every n times the timer B2 underflows. (n = setting value of the ICTB2 counter)

Figure 16.6 ICTB2 Register

16.1 Modulation Modes of Three-phase Motor Control Timers

The three-phase motor control timers supports two modulation modes: triangular wave modulation mode and sawtooth wave modulation mode. The triangular wave modulation mode has three-phase mode 0 and three-phase mode 1. Table 16.2 lists bit settings and characteristics of each mode.

Table 16.2 Modulation Modes

Item	Triangular Wave Modulation Mode		Sawtooth Wave Modulation Mode
	Three-phase mode 0	Three-phase mode 1	(Three-phase mode 0)
Bit settings	INV06 = 0, INV11 = 0, PWCON = 0	INV06 = 0, INV11 = 1	INV06 = 1, INV11 = 0, PWCON = 0
Waveform	Triangular wave		Sawtooth wave
Registers TA11, TA21, and TA41	Not used	Used	Not used
Timing to transfer data from registers IDB0 and IDB1 to the three-phase output shift register	Only once when a transfer trigger (1) occurs after setting registers IDB0 and IDB1		Whenever a transfer trigger (1) occurs
Timing to trigger the dead time timer when INV16 = 0	On the falling edge of a one-shot pulse of timers A1, A2, and A4		When a transfer trigger occurs, or on the falling edge of a one-shot pulse of timers A1, A2, and A4
Bits INV00 and INV01 in the INVC0 register	Disabled. The ICTB2 counter increments every time the timer B2 underflows, irrespective of the INV00 and INV01 bit settings	Enabled	Disabled. The ICTB2 counter increments every time the timer B2 underflows, irrespective of the INV00 and INV01 bit settings
INV13 bit	Disabled	Enabled	Disabled

Note:
1. Transfer trigger: an underflow of timer B2 and a write to the INV07 bit, or a write to the TB2 register when the INV10 bit is set to 1.

16.2 Timer B2

Timer B2, which operates in timer mode, is used for carrier wave control in the three-phase motor control timers.

Figure 16.7 and Figure 16.8 show registers TB2 and TB2MR in this function, respectively. Figure 16.9 shows the TB2SC register which switches timing to change the carrier wave frequency in three-phase mode 1.

Timer B2 Register (1)

b15 b8 b7 b0

Symbol	Address	Reset Value
TB2	0355h-0354h	Undefined

Function	Setting Range	RW
Divides the count source by n+1. Starts timers A1, A2, and A4 every time an underflow occurs (n = setting value)	0000h to FFFFh	RW

Note:
1. A 16-bit read/write access to this register should be performed.

Figure 16.7 TB2 Register in Three-phase Motor Control Timers

Timer B2 Mode Register

b7 b6 b5 b4 b3 b2 b1 b0

0 X 0 0 0 0

Symbol	Address	Reset Value
TB2MR	035Dh	00XX 0000b

Bit Symbol	Bit Name	Function	RW
TMOD0	Operating Mode Select Bit	Should be written with 00b (timer mode) when using the three-phase motor control timers	RW
TMOD1			RW
MR0	Disabled in the three-phase motor control timers. Should be written with 0 and read as undefined value		RW
MR1			RW
MR2	No register bit; should be written with 0 and read as undefined value		—
MR3	Disabled in the three-phase motor control timers. Should be written with 0 and read as undefined value		—
TCK0	Count Source Select Bit	b7 b6 0 0 : f1 0 1 : f8 1 0 : f2n 1 1 : fC32	RW
TCK1			RW

Figure 16.8 TB2MR Register in Three-phase Motor Control Timers

Timer B2 Special Mode Register

b7	b6	b5	b4	b3	b2	b1	b0
X	X	X	X	X	X	X	

Symbol	Address	Reset Value
TB2SC	035Eh	XXXX XXX0b

Bit Symbol	Bit Name	Function	RW
PWCON	Timer B2 Reload Timing Switching Bit (1)	0: The underflow of timer B2 1: The underflow of timer B2 when the reload control signal for timer A1 is 0	RW
— (b7-b1)	No register bits; should be written with 0 and read as undefined value		—

Note:
1. When the INV11 bit is set to 0 (three-phase mode 0) or the INV06 bit is set to 1 (sawtooth wave modulation mode), this bit should be set to 0.

Figure 16.9 TB2SC Register

16.3 Timers A4, A1, and A2

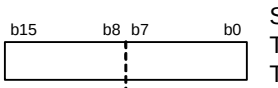
Timers A4, A1, and A2 are used for three-phase PWM output (U, $\overline{U}$, V, $\overline{V}$, W, and $\overline{W}$) control in the three-phase motor control timers.

These timers should operate in one-shot timer mode. Every time the timer B2 underflows, a trigger is input to timers A4, A1, and A2 to generate a one-shot pulse. If the values of registers TA4, TA1 and TA2 are rewritten every time a timer B2 interrupt is generated, the duty ratio of the PWM waveform can be varied.

In three-phase mode 1, the value of registers TAi and TAi-1 (i = 4, 1, 2) is alternately reloaded to the counter on each timer B2 interrupt, which halves timer B2 interrupt frequency. The sum of setting values for registers TAi and TAi1 should be identical to the setting value of the TB2 register in this mode.

Figure 16.10 shows registers TA1, TA2, TA4, TA11, TA21, and TA41 in the three-phase motor control timers. Figure 16.11 shows registers TA1MR, TA2MR, and TA4MR in this function. Figure 16.12 and Figure 16.13 show registers TRGSR and TABSR, respectively, in this function.

Timer Ai/Timer Ai-1 Registers (i = 1, 2, 4) (1 to 6)



Symbol	Address	Reset Value
TA1, TA2, TA4	0349h-0348h, 034Bh-034Ah, 034Fh-034Eh	Undefined
TA11, TA21, TA41	0303h-0302h, 0305h-0304h, 0307h-0306h	Undefined

Function	Setting Range	RW
The timer stops when the nth count source is counted after a start trigger is generated. Output signal for each phase is switched when timers A1, A2, and A4 stop (n = setting value)	0000h to FFFFh	WO

Notes:
1. A 16-bit write access to these registers should be performed.
2. When these registers are set to 0000h, the counter does not start, and no timer Ai interrupt request is generated.
3. The MOV instruction should be used to set these registers.
4. When the INV15 bit in the INVC1 register is set to 0 (dead time enabled), the turn-on output signal is switched to its active state with a delay. It is switched when the dead time timer stops.
5. When the INV11 bit in the INVC1 register is set to 0 (three-phase mode 0), the value of the TAi register is transferred to the reload register by a timer Ai start trigger. When the INV11 bit is set to 1 (three-phase mode 1), firstly the value of the TAi1 register is transferred to the reload register by a timer Ai start trigger. Then the value of the TAi register is transferred by the next timer Ai start trigger. After that the value of these registers TAi1 and TAi is alternately transferred to the reload register.
6. These registers should not be written when the timer B2 underflows.

Figure 16.10 Registers TA1, TA2, TA4, TA11, TA21, and TA41

Timer Ai Mode Register (i = 1, 2, 4)

b7	b6	b5	b4	b3	b2	b1	b0
		0	1	0	0	1	0

Symbol	Address	Reset Value
TA1MR, TA2MR, TA4MR	0357h, 0358h, 035Ah	0000 0000b

Bit Symbol	Bit Name	Function	RW
TMOD0	Operating Mode Select Bit	Should be written with 10b (one-shot timer mode) in three-phase motor control timers	RW
TMOD1			RW
MR0	Reserved	Should be written with 0	RW
MR1	External Trigger Select Bit	Should be written with 0 in three-phase motor control timers	RW
MR2	Trigger Select Bit	Should be written with 1 (selected by the TRGSR register) in three-phase motor control timers	RW
MR3	Should be written with 0 in three-phase motor control timers		RW
TCK0	Count Source Select Bit	b7 b6 0 0 : f1 0 1 : f8 1 0 : f2n 1 1 : fC32	RW
TCK1			RW

Figure 16.11 Registers TA1MR, TA2MR, and TA4MR in Three-phase Motor Control Timers

Trigger Select Register

b7 b6 b5 b4 b3 b2 b1 b0

Symbol	Address	Reset Value
TRGSR	0343h	0000 0000b

Bit Symbol	Bit Name	Function	RW
TA1TGL	Timer A1 Event/Trigger Select Bit	Should be set to 01b (the underflow of TB2) to use V-phase output control circuit	RW
TA1TGH			RW
TA2TGL	Timer A2 Event/Trigger Select Bit	Should be set to 01b (the underflow of TB2) to use W-phase output control circuit	RW
TA2TGH			RW
TA3TGL	Timer A3 Event/Trigger Select Bit	b5 b4 0 0 : Select the input to the TA3IN pin 0 1 : Select the overflow of TB2 (1) 1 0 : Select the overflow of TA2 (1) 1 1 : Select the overflow of TA4 (1)	RW
TA3TGH			RW
TA4TGL	Timer A4 Event/Trigger Select Bit	Should be set to 01b (the underflow of TB2) to use U-phase output control circuit	RW
TA4TGH			RW

Note:
1. The timer overflows or underflows.

Figure 16.12 TRGSR Register in Three-phase Motor Control Timers

Count Start Register

b7 b6 b5 b4 b3 b2 b1 b0

Symbol	Address	Reset Value
TABSR	0340h	0000 0000b

Bit Symbol	Bit Name	Function	RW
TA0S	Timer A0 Count Start Bit	0: Stop counter 1: Start counter	RW
TA1S	Timer A1 Count Start Bit	0: Stop counter 1: Start counter	RW
TA2S	Timer A2 Count Start Bit	0: Stop counter 1: Start counter	RW
TA3S	Timer A3 Count Start Bit	0: Stop counter 1: Start counter	RW
TA4S	Timer A4 Count Start Bit	0: Stop counter 1: Start counter	RW
TB0S	Timer B0 Count Start Bit	0: Stop counter 1: Start counter	RW
TB1S	Timer B1 Count Start Bit	0: Stop counter 1: Start counter	RW
TB2S	Timer B2 Count Start Bit	0: Stop counter 1: Start counter	RW

Figure 16.13 TABSR Register

16.4 Simultaneous Conduction Prevention and Dead Time Timer

The three-phase motor control timers offers two ways to avoid shoot-through, which occurs when high-side and low-side transistors are simultaneously turned on.

One is by the function called “simultaneous turn-on signal output disable function”. This function prevents high-side and low-side transistors from being inadvertently switched active caused by program errors and so on. The other is by the use of dead time timers. A dead time timer delays the turn-on of one transistor in order to ensure that an adequate time (the dead time) passes after the turn-off of the other.

To disable simultaneous turn-on output signals, the INV04 bit in the INVC0 register should be set to 1. If outputs for any pair of phases (U and $\overline{U}$, V and $\overline{V}$, or W and $\overline{W}$) are simultaneously switched to an active state, every three-phase motor control output pin becomes high-impedance. Figure 16.14 shows an example of output waveform when simultaneous turn-on signal output is disabled.

To enable the dead time timer, the INV15 bit in the INVC1 register should be set to 0. The DTT register determines the dead time. Figure 16.15 shows the DTT register and Figure 16.16 shows an example of output waveform on using dead time timer.

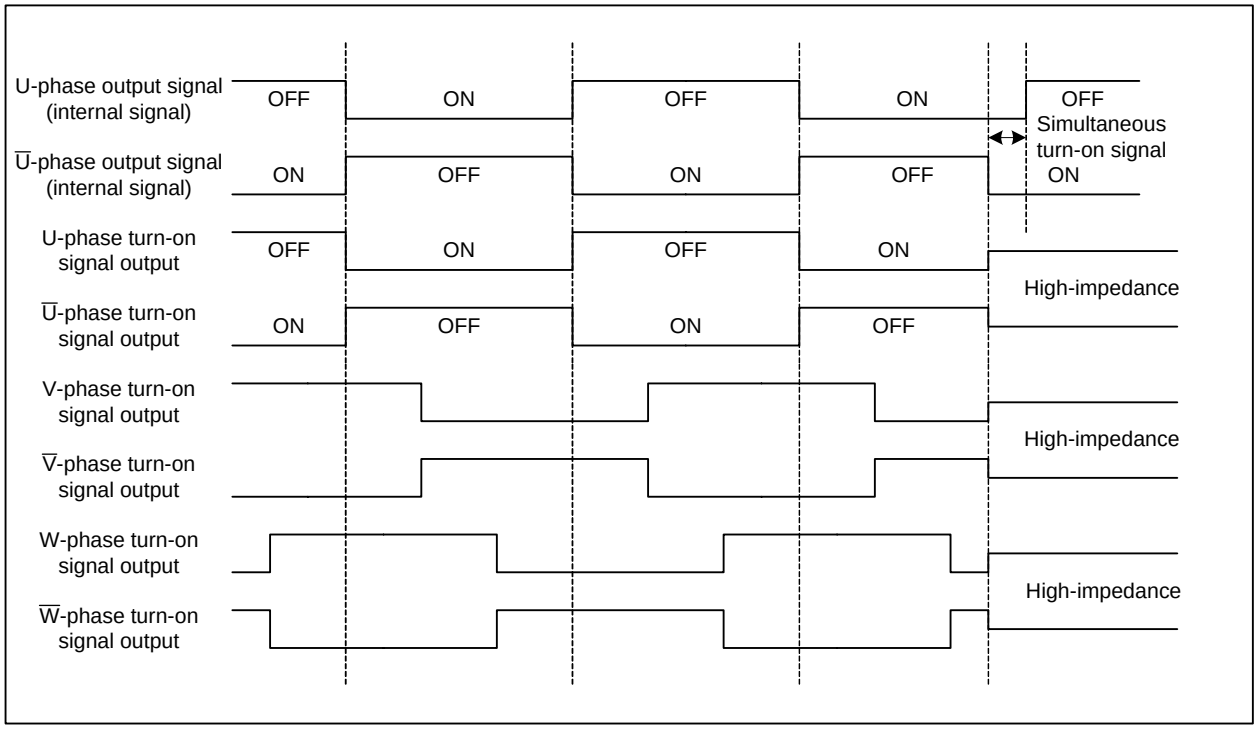


Figure 16.14 Output Waveform When Simultaneous Turn-on Signal Output is Disabled

Dead Time Timer (1, 2)

b7 b0

Symbol	Address	Reset Value
DTT	030Ch	Undefined

Function	Setting Range	RW
The dead time timer is a one-shot timer to delay the timing for a turn-on signal to be switched to its active state preventing a simultaneous conduction of high-side and low-side transistors. The timer stops when counting a count source n times after a start trigger occurs (n = setting value) (3)	01h to FFh	WO

Notes:
1. The MOV instruction should be used to set this register.
2. This register setting is enabled when the INV15 bit in the INVC1 register is set to 0 (dead time enabled). No dead time can be set when the INV15 bit is set to 1 (dead time disabled).
3. The trigger and count source should be respectively selected using bits INV06 and INV12 in the INVC1 register.

Figure 16.15 DTT Register

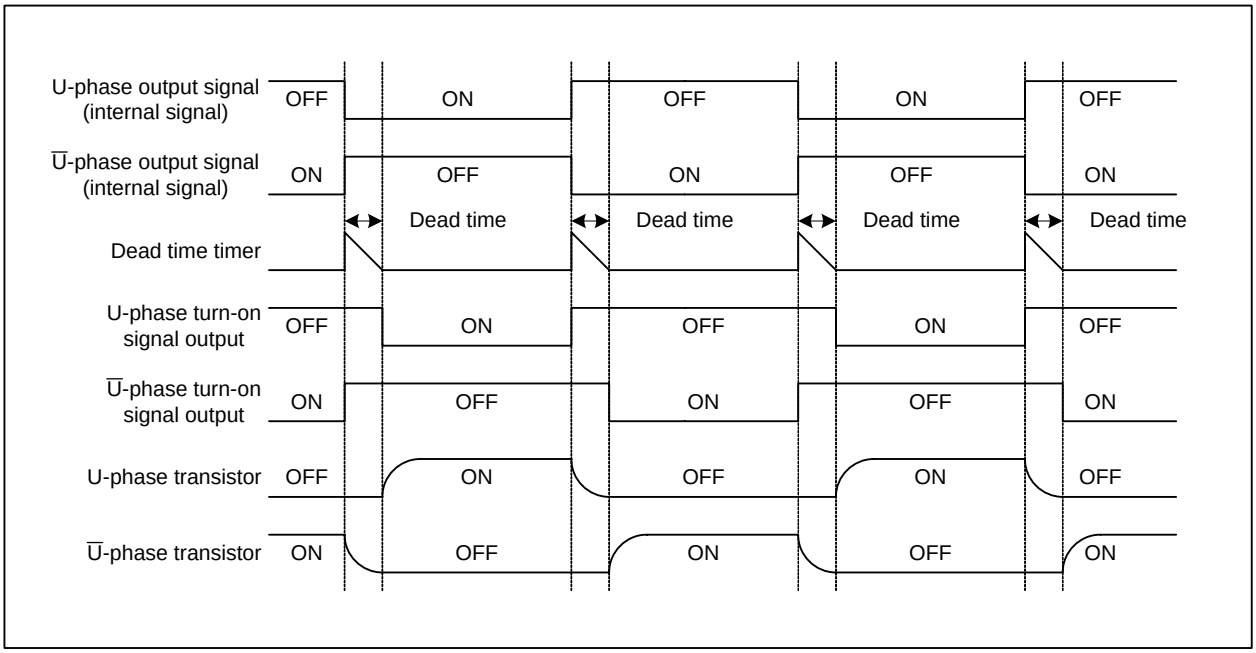
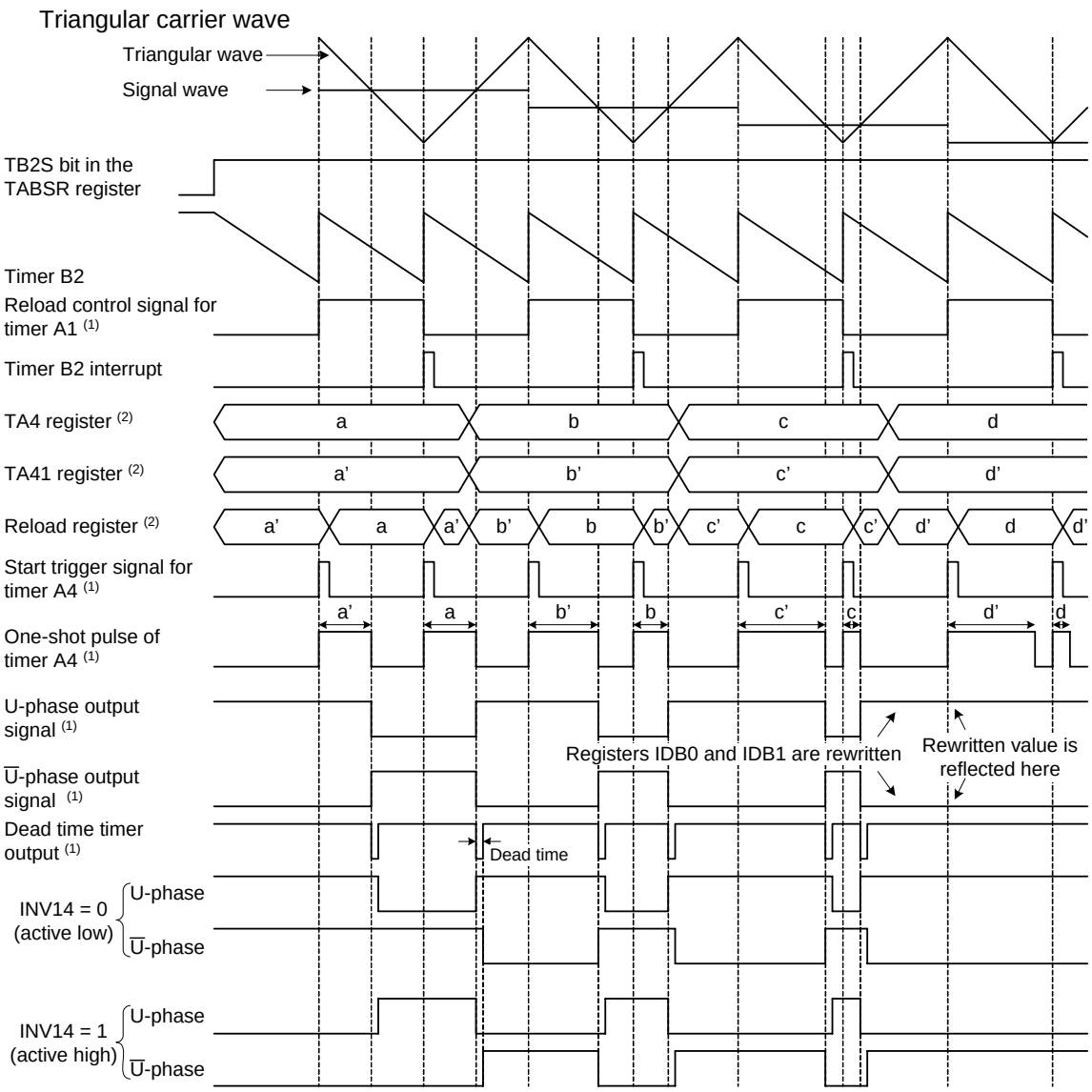


Figure 16.16 Output Waveform on Using Dead Time Timer

16.5 Three-phase Motor Control Timer Operation

Figure 16.17 and Figure 16.18 show an operation example of triangular wave modulation and sawtooth wave modulation, respectively.



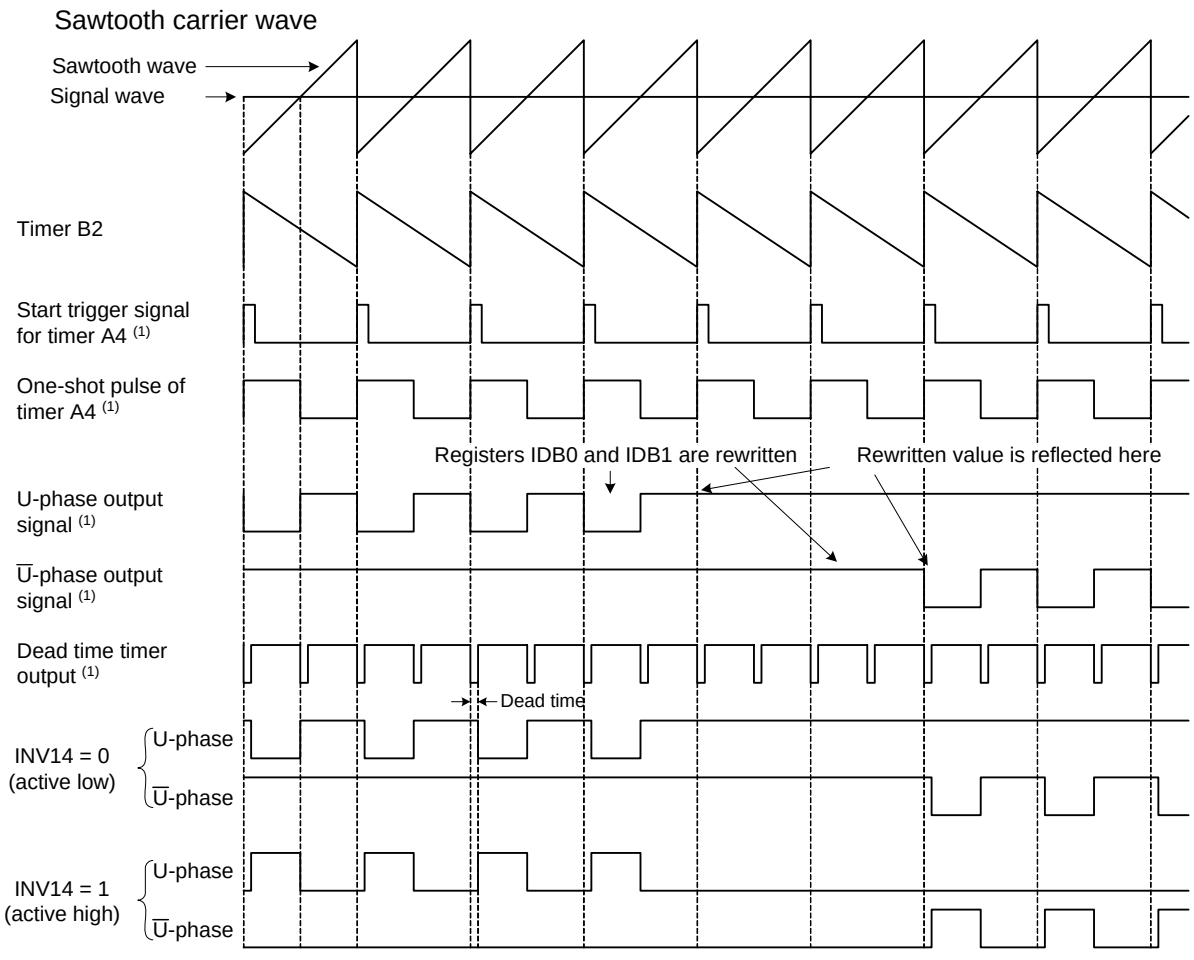
The above applies to INVC0 = 00XX11XXb (X varies depending on each system) and INVC1 = 010XXXX0b.
The PWM output may vary as follows:

Case 1: When INV11 = 1 (three-phase mode 1)
- -INV01 = 0 and ICTB2 = 2h (timer B2 interrupt generated every 2nd time the timer B2 underflows), or INV01 = 1, INV00 = 1, and ICTB2 = 1h (timer B2 interrupt generated every time the timer B2 underflows when the reload control signal for timer A1 is set to 1)
- -The setting of registers TA4 and TA41 are varied every time a timer B2 interrupt is generated,
  Default value: TA41 = a', TA4 = a
  On the first timer B2 interrupt: TA41 = b', TA4 = b; the second time: TA41 = c', TA4 = c
- -Default value of registers IDB0 and IDB1: DU0 = 1, DUB0 = 0, DU1 = 0, DUB1 = 1
  On the third time: DU0 = 1, DUB0 = 0, DU1 = 1, DUB1 = 0

Case 2: When INV11 = 0 (three-phase mode 0)
- -INV01 = 0 and ICTB2 = 1h (timer B2 interrupt generated every time timer B2 underflows)
- -TA4 register setting is varied every time a timer B2 interrupt is generated,
  Default value: TA4 = a'
  On the first timer B2 interrupt: TA4 = a; 2nd time: TA4 = b'; 3rd time: TA4 = b; 4th time: TA4 = c'; 5th time: TA4 = c
- -Default value of registers IDB0 and IDB1: DU0 = 1, DUB0 = 0, DU1 = 0, DUB1 = 1
  On the sixth time: DU0 = 1, DUB0 = 0, DU1 = 1, DUB1 = 0

Notes:
1. Internal signal. Refer to the block diagram of three-phase motor control timers.
2. Applied when the INV11 bit in the INV1 register is set to 1 (three-phase mode 1).

Figure 16.17 Triangular Wave Modulation Operation



The above applies to INVC0 = 01XX110Xb (X varies depending on each system) and INVC1 = 000XXX00b.
This bit setting is applicable to turn-on control with a phase shift of 120 degrees.

The PWM output may vary as follows:
Default value of registers IDB0 and IDB1: DU0 = 0, DUB0 = 1, DU1 = 1, DUB1 = 1
On the third timer B2 interrupt: DU0 = 1, DUB0 = 1, DU1 = 1, DUB1 = 1
On the fifth timer B2 interrupt: DU0 = 1, DUB0 = 0, DU1 = 1, DUB1 = 1

Note:
1. Internal signal. Refer to the block diagram of three-phase motor control timers.

Figure 16.18 Sawtooth Wave Modulation Operation

16.6 Notes on Three-phase Motor Control Timers

16.6.1 Shutdown

- When a low signal is applied to the $\overline{\text{NMI}}$ pin with the bit settings below, pins TA1OUT, TA2OUT, and TA4OUT become high-impedance: the PM24 bit in the PM2 register is 1 (NMI enabled), the INV02 bit in the INVC0 register is 1 (the three-phase motor control timers used), and the INV03 bit is 1 (the three-phase motor control timer output enabled).

16.6.2 Register setting

- Do not write to the TAi1 register (i = 1, 2, 4) in the timing that timer B2 underflows. Before writing to the TAi1 register, read the TB2 register to verify that sufficient time is left until timer B2 underflows. Then, immediately write to the TAi1 register so that no interrupt handler is performed during this write procedure. If the TB2 register indicates little time is left until the underflow, write to the TAi1 register after timer B2 underflows.

17. Serial Interface

Serial interface consists of nine channels (UART0 to UART8).
Each UARTi (i = 0 to 8) has an exclusive timer to generate the transmit/receive clock and operates independently.
Figure 17.1 and Figure 17.2 show respectively a block diagram of UART0 to UART6 and that of UART7 and UART8.
The UARTi supports following modes:

- Synchronous serial interface mode (for UART0 to UART8)
- Asynchronous serial interface mode (UART mode) (for UART0 to UART8)
- Special mode 1 (I^2C mode) (for UART0 to UART6)
- Special mode 2 (for UART0 to UART6)
- Special mode 4 (Bus collision detection: IE mode) (optional) (1) (for UART0 to UART6)

Figure 17.3 to Figure 17.19 show registers associated with the UARTi.
Refer to the tables listing each mode for registers and pin settings.

Note:
1. Contact a Renesas Electronics sales office to use the optional features.

Table 17.1 Functions of UART0 to UART8

Mode/Function	UART0 to UART6	UART7, UART8
Synchronous serial interface mode	Available	Available
Serial data logical inversion	Available	Not available
UART mode	Available	Available
CTS/RTS selection	Available	Available
TXD and RXD I/O polarity selection	Available	Not available
Special mode 1 (I^2C mode)	Available	Not available
Special mode 2	Available	Not available
Special mode 4 (IE mode) (optional) (1)	Available	Not available
Pins TXD and RXD output mode	Push-pull output, N-channel open drain output programmable by port function select registers	Push-pull output, N-channel open drain output programmable by port function select registers

Note:
1. Contact a Renesas Electronics sales office to use the optional features.

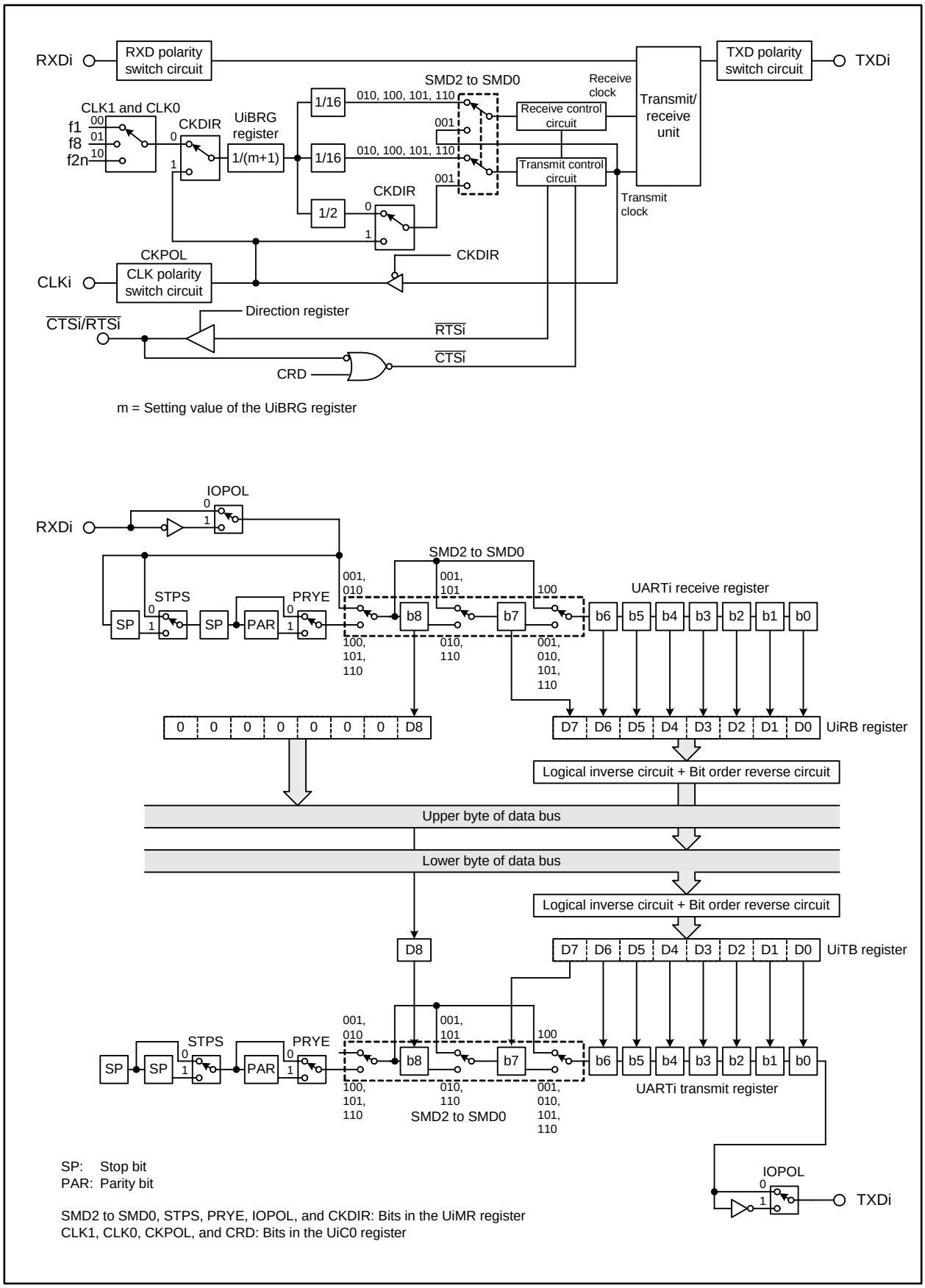


Figure 17.1 UARTi Block Diagram (i = 0 to 6)

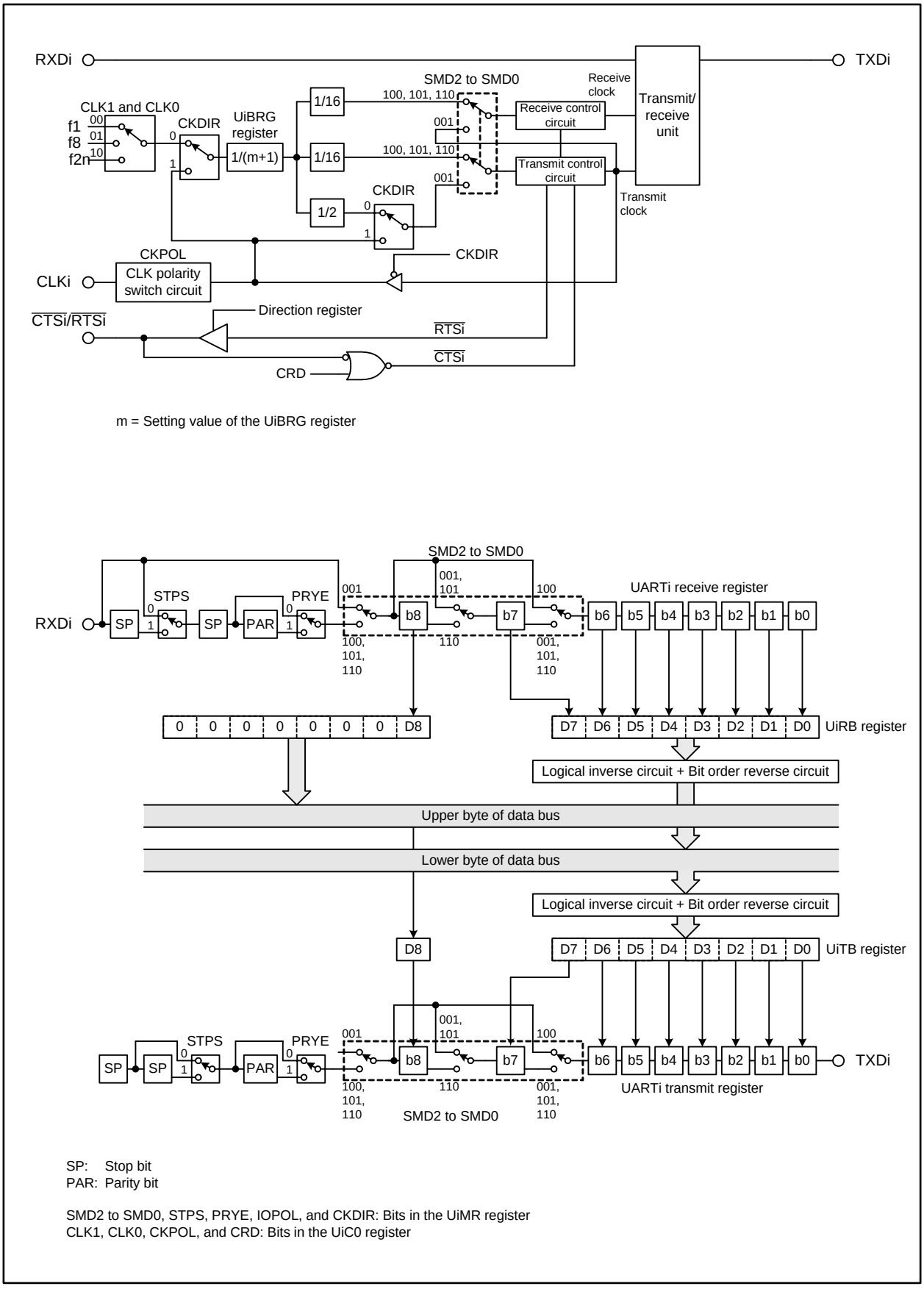
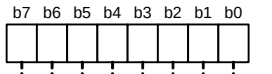


Figure 17.2 UARTi Block Diagram (i = 7, 8)

UARTi Transmit/Receive Mode Register (i = 0 to 6)



Symbol	Address	Reset Value
U0MR to U3MR	0368h, 02E8h, 0338h, 0328h	0000 0000b
U4MR to U6MR	02F8h, 01C8h, 01D8h	0000 0000b

Bit Symbol	Bit Name	Function	RW
SMD0	Serial Interface Mode Select Bit	b2 b1 b0 0 0 0 : Serial interface disabled 0 0 1 : Synchronous serial interface mode 0 1 0 : I^2C mode 1 0 0 : UART mode, 7-bit character length 1 0 1 : UART mode, 8-bit character length 1 1 0 : UART mode, 9-bit character length Only use the combinations listed above	RW
SMD1			RW
SMD2			RW
CKDIR	Internal/External Clock Select Bit	0: Internal clock 1: External clock	RW
STPS	Stop Bit Length Select Bit	0: 1 stop bit 1: 2 stop bits	RW
PRY	Odd/Even Parity Select Bit	Enabled when the PRYE bit is 1 0: Odd parity 1: Even parity	RW
PRYE	Parity Enable Bit	0: Parity disabled 1: Parity enabled	RW
IOPOL	TXD, RXD Input/Output Polarity Switch Bit	0: Non inverted 1: Inverted	RW

Figure 17.3 Registers U0MR to U6MR

UARTi Transmit/Receive Mode Register (i = 7, 8)

b7	b6	b5	b4	b3	b2	b1	b0
0							

Symbol	Address	Reset Value
U7MR, U8MR	01E0h, 01E8h	0000 0000b

Bit Symbol	Bit Name	Function	RW
SMD0	Serial Interface Mode Select Bit	b2 b1 b0 0 0 0 : Serial interface disabled 0 0 1 : Synchronous serial interface mode 1 0 0 : UART mode, 7-bit character length 1 0 1 : UART mode, 8-bit character length 1 1 0 : UART mode, 9-bit character length Only use the combinations listed above	RW
SMD1			RW
SMD2			RW
CKDIR	Internal/External Clock Select Bit	0: Internal clock 1: External clock	RW
STPS	Stop Bit Length Select Bit	0: 1 stop bit 1: 2 stop bits	RW
PRY	Odd/Even Parity Select Bit	Enabled when the PRYE bit is 1 0: Odd parity 1: Even parity	RW
PRYE	Parity Enable Bit	0: Parity disabled 1: Parity enabled	RW
— (b7)	Reserved	Should be written with 0	RW

Figure 17.4 Registers U7MR and U8MR

UARTi Transmit/Receive Control Register 0 (i = 0 to 6)

b7	b6	b5	b4	b3	b2	b1	b0
		0			0		

Symbol	Address	Reset Value
U0C0 to U3C0	036Ch, 02ECh, 033Ch, 032Ch	0000 1000b
U4C0 to U6C0	02FCh, 01CCh, 01DCh	0000 1000b

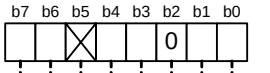
Bit Symbol	Bit Name	Function	RW
CLK0	UiBRG Count Source Select Bit	b1 b0 0 0 : f1 0 1 : f8 1 0 : f2n 1 1 : Do not use this combination	RW
CLK1			RW
— (b2)	Reserved	Should be written with 0	RW
TXEPT	Transmit Shift Register Empty Flag	0: Data held in the transmit shift register (transmission in progress) 1: No data held in the transmit shift register (transmission completed)	RO
CRD	CTS Disable Bit	0: CTS enabled 1: CTS disabled	RW
— (b5)	Reserved	Should be written with 0	RW
CKPOL	CLK Polarity Select Bit	0: Output transmit data on the falling edge of the transmit/receive clock and input receive data on the rising edge 1: Output transmit data on the rising edge of the transmit/receive clock and input receive data on the falling edge	RW
UFORM	Bit Order Select Bit (1)	0: LSB first 1: MSB first	RW

Note:

1. This bit is enabled when bits SMD2 to SMD0 in the UiMR register are set to 001b (synchronous serial interface mode selected) or 101b (UART mode, 8-bit character length selected). It should be set to 1 when bits SMD2 to SMD0 are set to 010b (I^2C mode selected) and should be set to 0 when they are set to 100b (UART mode, 7-bit character length selected) or 110b (UART mode, 9-bit character length selected).

Figure 17.5 Registers U0C0 to U6C0

UARTi Transmit/Receive Control Register 0 (i = 7, 8)



Symbol	Address	Reset Value
U7C0, U8C0	01E4h, 01ECh	00X0 1000b

Bit Symbol	Bit Name	Function	RW
CLK0	UiBRG Count Source Select Bit	b1 b0 0 0 : f1 0 1 : f8 1 0 : f2n 1 1 : Do not use this combination	RW
CLK1			RW
— (b2)	Reserved	Should be written with 0	RW
TXEPT	Transmit Shift Register Empty Flag	0: Data held in the transmit shift register (transmission in progress) 1: No data held in the transmit shift register (transmission completed)	RO
CRD	CTS Disable Bit	0: CTS enabled 1: CTS disabled	RW
— (b5)	No registers bit; should be written with 0 and read as undefined value		—
CKPOL	CLK Polarity Select Bit	0: Output transmit data on the falling edge of the transmit/receive clock and input receive data on the rising edge 1: Output transmit data on the rising edge of the transmit/receive clock and input receive data on the falling edge	RW
UFORM	Bit Order Select Bit (1)	0: LSB first 1: MSB first	RW

Note:
1. This bit is enabled when bits SMD2 to SMD0 in the UiMR register are set to 001b (synchronous serial interface mode selected) or 101b (UART mode, 8-bit character length selected). It should be set to 0 when they are set to 100b (UART mode, 7-bit character length selected) or 110b (UART mode, 9-bit character length selected).

Figure 17.6 Registers U7C0 and U8C0

UARTi Transmit/Receive Control Register 1 (i = 0 to 6)

b7 b6 b5 b4 b3 b2 b1 b0

0

Symbol	Address	Reset Value
U0C1 to U3C1	036Dh, 02EDh, 033Dh, 032Dh	0000 0010b
U4C1 to U6C1	02FDh, 01CDh, 01DDh	0000 0010b

Bit Symbol	Bit Name	Function	RW
TE	Transmit Enable Bit	0: Transmission disabled 1: Transmission enabled	RW
TI	Transmit Buffer Empty Flag	0: Data held in the UiTB register 1: No data held in the UiTB register	RO
RE	Receive Enable Bit	0: Reception disabled 1: Reception enabled	RW
RI	Receive Complete Flag	0: No data held in the UiRB register 1: Data held in the UiRB register	RO
UiIRS	UARTi Transmit Interrupt Source Select Bit	0: UiTB register is empty (TI = 1) 1: Transmission is completed (TXEPT = 1)	RW
UiRRM	UARTi Continuous Receive Mode Enable Bit	0: Continuous receive mode disabled 1: Continuous receive mode enabled	RW
UiLCH	Logical Inversion Select Bit (1)	0: Data non logic-inverted 1: Data logic-inverted	RW
— (b7)	Reserved	Should be written with 0	RW

Note:

1. This bit is enabled when bits SMD2 to SMD0 in the UiMR register are set to 001b (synchronous serial interface mode selected), 100b (UART mode, 7-bit character length selected), or 101b (UART mode, 8-bit character length selected). It should be set to 0 when bits SMD2 to SMD0 are set to 010b (I^2C mode selected) or 110b (UART mode, 9-bit character length selected).

Figure 17.7 Registers U0C1 to U6C1

UARTi Transmit/Receive Control Register 1 (i = 7, 8)

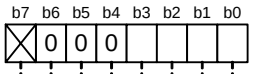
b7 b6 b5 b4 b3 b2 b1 b0

Symbol	Address	Reset Value
U7C1, U8C1	01E5h, 01EDh	XXXX 0010b

Bit Symbol	Bit Name	Function	RW
TE	Transmit Enable Bit	0: Transmission disabled 1: Transmission enabled	RW
TI	Transmit Buffer Empty Flag	0: Data held in the UiTB register 1: No data held in the UiTB register	RO
RE	Receive Enable Bit	0: Reception disabled 1: Reception enabled	RW
RI	Receive Complete Flag	0: No data held in the UiRB register 1: Data held in the UiRB register	RO
— (b7-b4)	No register bits; should be written with 0 and read as undefined value		—

Figure 17.8 Registers U7C1 and U8C1

UART7, UART8 Transmit/Receive Control Register 2

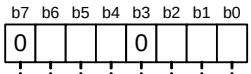


Symbol: U78CON
Address: 01F0h
Reset Value: X000 0000b

Bit Symbol	Bit Name	Function	RW
U7IRS	UART7 Transmit Interrupt Source Select Bit	0: Transmit buffer is empty (TI = 1) 1: Transmission is completed (TXEPT = 1)	RW
U8IRS	UART8 Transmit Interrupt Source Select Bit	0: Transmit buffer is empty (TI = 1) 1: Transmission is completed (TXEPT = 1)	RW
U7RRM	UART7 Continuous Receive Mode Enable Bit	0: Continuous receive mode disabled 1: Continuous receive mode enabled	RW
U8RRM	UART8 Continuous Receive Mode Enable Bit	0: Continuous receive mode disabled 1: Continuous receive mode enabled	RW
— (b6-b4)	Reserved	Should be written with 0	RW
— (b7)	No register bit; should be written with 0 and read as undefined value		—

Figure 17.9 U78CON Register

UARTi Special Mode Register (i = 0 to 6)



Symbol	Address	Reset Value
U0SMR to U3SMR	0367h, 02E7h, 0337h, 0327h	0000 0000b
U4SMR to U6SMR	02F7h, 01C7h, 01D7h	0000 0000b

Bit Symbol	Bit Name	Function	RW
IICM	I^2C Mode Select Bit (1)	0: Non-I^2C mode 1: I^2C mode	RW
ABC	Arbitration Lost Detection Flag Control (1)	0: Update every bit 1: Update every byte	RW
BBS	Bus Busy Flag (1, 2)	0: Detect stop condition 1: Detect start condition (bus busy)	RW
— (b3)	Reserved	Should be written with 0	RW
ABSCS	Bus Collision Detect Sampling Clock Select Bit (3)	0: Rising edge of the transmit/receive clock 1: The underflow of timer Aj (j = 0, 3, 4) (4)	RW
ACSE	Transmit Enable Bit Auto-reset to Zero Select Bit (3)	0: Non auto-reset to zero 1: Auto-reset to zero at bus collision	RW
SSS	Transmit Start Condition Select Bit (3)	0: No relation with RXDi 1: Synchronized with RXDi	RW
— (b7)	Reserved	Should be written with 0	RW

Notes:
1. This bit is used in I^2C mode.
2. The BBS bit is only set to 0. This setting is unchanged even if it is set to 1.
3. This bit is used in IE mode.
4. UART0: timer A3 underflow signal, UART1: timer A4 underflow signal
   UART2: timer A0 underflow signal, UART3: timer A3 underflow signal
   UART4: timer A4 underflow signal, UART5: timer A3 underflow signal
   UART6: timer A4 underflow signal

Figure 17.10 Registers U0SMR to U6SMR

UARTi Special Mode Register 2 (i = 0 to 6)

b7	b6	b5	b4	b3	b2	b1	b0
0							

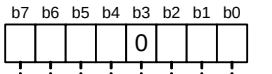
Symbol	Address	Reset Value
U0SMR2 to U3SMR2	0366h, 02E6h, 0336h, 0326h	0000 0000b
U4SMR2 to U6SMR2	02F6h, 01C6h, 01D6h	0000 0000b

Bit Symbol	Bit Name	Function	RW
IICM2	I^2C Mode Select Bit 2	0: Use ACK/NACK interrupt 1: Use transmit/receive interrupt	RW
CSC	Clock Synchronous Bit (1)	0: Clock synchronization disabled 1: Clock synchronization enabled	RW
SWC	SCL Wait Auto Insert Bit (2)	0: No wait-state/wait-state cleared 1: Hold the SCLi pin low after the eighth bit is received	RW
ALS	SDA Output Auto Stop Bit (1)	When arbitration lost is detected, 0: Do not stop the SDAi output 1: Stop the SDAi output	RW
STC	UARTi Auto Initialize Bit (2)	When the start condition is detected, 0: Do not initialize the circuit 1: Initialize the circuit	RW
SWC2	SCL Wait Output Bit 2 (1)	0: Output the transmit/receive clock at the SCLi pin 1: Hold the SCLi pin low	RW
SDHI	SDA Output Stop Bit (2)	0: Output data 1: Stop the output (high-impedance)	RW
— (b7)	Reserved	Should be written with 0	RW

Notes:
1. This bit is used in master mode of I^2C mode.
2. This bit is used in slave mode of I^2C mode.

Figure 17.11 Registers U0SMR2 to U6SMR2

UARTi Special Mode Register 3 (i = 0 to 6)



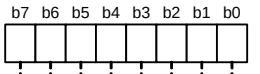
Symbol	Address	Reset Value
U0SMR3 to U3SMR3	0365h, 02E5h, 0335h, 0325h	0000 0000b
U4SMR3 to U6SMR3	02F5h, 01C5h, 01D5h	0000 0000b

Bit Symbol	Bit Name	Function	RW
SSE	$\overline{SS}$ Pin Function Enable Bit (1, 2)	0: SS disabled 1: SS enabled	RW
CKPH	Clock-phase Set Bit	0: No clock delay 1: Clock delayed	RW
DINC	Serial Input Pin Set Bit (1)	0: Select the TXDi/RXDi pin (master mode) 1: Select the STXDi/SRXDi pin (slave mode)	RW
— (b3)	Reserved	Should be written with 0	RW
ERR	Mode Fault Flag (1)	0: No error detected 1: Error detected (3)	RW
DL0	SDAi Digital Delay Time Set Bit (4, 5)	Based on the UiBRG count source, the output at the SDAi pin is delayed as below: b7 b6 b5 0 0 0 : No delay 0 0 1 : 1 to 2 cycles 0 1 0 : 2 to 3 cycles 0 1 1 : 3 to 4 cycles 1 0 0 : 4 to 5 cycles 1 0 1 : 5 to 6 cycles 1 1 0 : 6 to 7 cycles 1 1 1 : 7 to 8 cycles	RW
DL1			RW
DL2			RW

Notes:
1. This bit is used in special mode 2.
2. To use the SS function, the CRD bit in the UiC0 register should be set to 1 (CTS disabled).
3. The ERR bit is only set to 0. This setting is unchanged even if it is set to 1.
4. Bits DL2 to DL0 in I^2C mode generate a digital delay for the output at the SDAi pin. These bits should be set to 000b (no delay) in all modes other than I^2C mode.
5. When the external clock is selected, a delay of approximately 100 ns is added.

Figure 17.12 Registers U0SMR3 to U6SMR3

UARTi Special Mode Register 4 (i = 0 to 6)



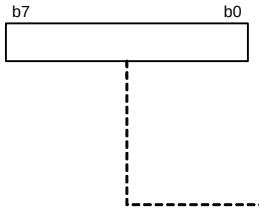
Symbol	Address	Reset Value
U0SMR4 to U3SMR4	0364h, 02E4h, 0334h, 0324h	0000 0000b
U4SMR4 to U6SMR4	02F4h, 01C4h, 01D4h	0000 0000b

Bit Symbol	Bit Name	Function	RW
STAREQ	Start Condition Generate Bit (1)	0: Clear 1: Start (2)	RW
RSTAREQ	Restart Condition Generate Bit (1)	0: Clear 1: Start (2)	RW
STPREQ	Stop Condition Generate Bit (1)	0: Clear 1: Start (2)	RW
STSPSEL	SCL, SDA Output Select Bit (1)	0: Select serial I/O circuit 1: Select start condition/stop condition generate circuit (3)	RW
ACKD	ACK Data Bit (4)	0: ACK 1: NACK	RW
ACKC	ACK Data Output Enable Bit (4)	0: Serial data output 1: ACK data output	RW
SCLHI	SCL Output Stop Bit (1)	When stop condition is detected, 0: Do not stop SCLi output 1: Stop SCLi output	RW
SWC9	SCL Wait Auto Insert Bit 3 (4)	0: No wait-state/wait-state cleared 1: Hold the SCLi pin low after the ninth bit is received	RW

Notes:
1. This bit is used in master mode of I^2C mode. To set this bit to 1, preset the IICM bit in the UiSMR register to 1 (I^2C mode).
2. This bit becomes 0 when its condition is generated. The setting remains 1 when the condition is uncompleted.
3. The STSPSEL bit should be set to 1 after setting the STAREQ, RSTAREQ, or STPREQ bit to 1.
4. This bit is used in slave mode of I^2C mode. To set this bit to 1, preset the IICM bit in the UiSMR register to 1 (I^2C mode).

Figure 17.13 Registers U0SMR4 to U6SMR4

UARTi Bit Rate Register (i = 0 to 8) (1, 2, 3)



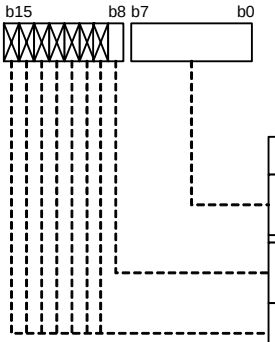
Symbol	Address	Reset Value
U0BRG to U3BRG	0369h, 02E9h, 0339h, 0329h	Undefined
U4BRG to U7BRG	02F9h, 01C9h, 01D9h, 01E1h	Undefined
U8BRG	01E9h	Undefined

Function	Setting Range	RW
The UiBRG register divides the count source by n+1 (n = setting value)	00h to FFh	WO

Notes:
1. Set bits CLK1 and CLK0 in the UiC0 register before rewriting this register.
2. The MOV instruction should be used to set this register.
3. The register should be set while no data is being transmitted/received.

Figure 17.14 Registers U0BRG to U8BRG

UARTi Transmit Buffer Register (i = 0 to 8) (1)



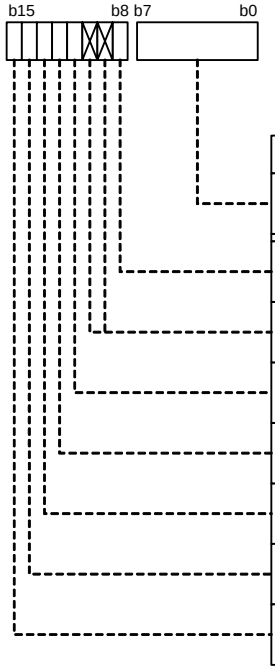
Symbol	Address	Reset Value
U0TB to U2TB	036Bh-036Ah, 02EBh-02EAh, 033Bh-033Ah	Undefined
U3TB to U5TB	032Bh-032Ah, 02FBh-02FAh, 01CBh-01CAh	Undefined
U6TB to U8TB	01DBh-01DAh, 01E3h-01E2h, 01EBh-01EAh	Undefined

Bit Symbol	Function	RW
— (b7-b0)	Data (D7 to D0) transmitted	WO
— (b8)	Data (D8) transmitted	WO
— (b15-b9)	No register bits; should be written with 0	—

Note:
1.The MOV instruction should be used to set this register.

Figure 17.15 Registers U0TB to U8TB

UARTi Receive Buffer Register (i = 0 to 6)



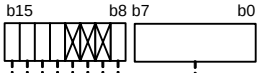
Symbol	Address	Reset Value
U0RB to U2RB	036Fh-036Eh, 02EFh-02EEh, 033Fh-033Eh	Undefined
U3RB to U5RB	032Fh-032Eh, 02FFh-02FEh, 01CFh-01CEh	Undefined
U6RB	01DFh-01DEh	Undefined

Bit Symbol	Bit Name	Function	RW
— (b7-b0)	—	Data (D7 to D0) received	RO
— (b8)	—	Data (D8) received	RO
— (b10-b9)	No register bits; should be written with 0 and read as 0		—
ABT	Arbitration Lost Detection Flag (1)	0: Not detected (win) 1: Detected (lose)	RW
OER	Overrun Error Flag (2)	0: No overrun error occurred 1: Overrun error occurred	RO
FER	Framing Error Flag (2, 3)	0: No framing error occurred 1: Framing error occurred	RO
PER	Parity Error Flag (2, 3)	0: No parity error occurred 1: Parity error occurred	RO
SUM	Error Sum Flag (2, 3)	0: No error occurred 1: Error occurred	RO

Notes:
1. The ABT bit can be set to 0 only.
2. Bits OER, FER, PER, and SUM are set to 0 when bits SMD2 to SMD0 in the UiMR register are set to 000b (serial interface disabled) or the RE bit in the UiC1 register is set to 0 (reception disabled). When bits OER, FER, and PER are all set to 0, the SUM bit is also set to 0. Bits FER and PER are set to 0 when the lower byte in the UiRB register is read.
3. When bits SMD2 to SMD0 are set to 001b (synchronous serial interface mode selected) or 010b (I^2C mode selected), these error flags are disabled and read as an undefined value.

Figure 17.16 Registers U0RB to U6RB

UARTi Receive Buffer Register (i = 7, 8)



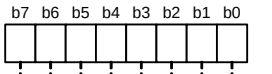
Symbol	Address	Reset Value
U7RB, U8RB	01E7h-01E6h, 01EFh-01EEh	Undefined

Bit Symbol	Bit Name	Function	RW
— (b7-b0)	—	Data (D7 to D0) received	RO
— (b8)	—	Data (D8) received	RO
— (b11-b9)	No register bits; should be written with 0 and read as 0		—
OER	Overrun Error Flag (1)	0: No overrun error occurred 1: Overrun error occurred	RO
FER	Framing Error Flag (1, 2)	0: No framing error occurred 1: Framing error occurred	RO
PER	Parity Error Flag (1, 2)	0: No parity error occurred 1: Parity error occurred	RO
SUM	Error Sum Flag (1, 2)	0: No error occurred 1: Error occurred	RO

Notes:
1. Bits OER, FER, PER, and SUM are set to 0 when bits SMD2 to SMD0 in the UiMR register are set to 000b (serial interface disabled) or the RE bit in the UiC1 register is set to 0 (reception disabled). When bits OER, FER, and PER are all set to 0, the SUM bit is also set to 0. Bits FER and PER are set to 0 when the lower byte in the UiRB register is read.
2. When bits SMD2 to SMD0 are set to 001b (synchronous serial interface mode selected) or 010b (I^2C mode selected), these error flags are disabled and read as an undefined value.

Figure 17.17 Registers U7RB and U8RB

External Interrupt Request Source Select Register 0



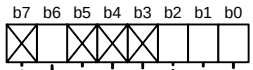
Symbol	Address	Reset Value
IFSR0	4406Fh	0000 0000b

Bit Symbol	Bit Name	Function	RW
IFSR00	$\overline{\text{INT0}}$ Pin Polarity Select Bit (1)	0: One edge 1: Both edges	RW
IFSR01	$\overline{\text{INT1}}$ Pin Polarity Select Bit (1)	0: One edge 1: Both edges	RW
IFSR02	$\overline{\text{INT2}}$ Pin Polarity Select Bit (1)	0: One edge 1: Both edges	RW
IFSR03	$\overline{\text{INT3}}$ Pin Polarity Select Bit (1)	0: One edge 1: Both edges	RW
IFSR04	$\overline{\text{INT4}}$ Pin Polarity Select Bit (1)	0: One edge 1: Both edges	RW
IFSR05	$\overline{\text{INT5}}$ Pin Polarity Select Bit (1)	0: One edge 1: Both edges	RW
IFSR06	UART0/UART3 Interrupt Source Select Bit	0: Bus collision, start condition detection, stop condition detection in UART3 1: Bus collision, start condition detection, stop condition detection in UART0	RW
IFSR07	UART1/UART4 Interrupt Source Select Bit	0: Bus collision, start condition detection, stop condition detection in UART4 1: Bus collision, start condition detection, stop condition detection in UART1	RW

Note:
1. This bit should be set to 0 to select the level sensitive input as trigger. To set this bit to 1, the POL bit in the corresponding INTiIC register (i = 0 to 5) should be set to 0 (falling edge).

Figure 17.18 IFSR0 Register

External Interrupt Request Source Select Register 1



Symbol	Address	Reset Value
IFSR1	4406Dh	X0XX X000b

Bit Symbol	Bit Name	Function	RW
IFSR10	$\overline{\text{INT6}}$ Pin Polarity Select Bit (1)	0: One edge 1: Both edges	RW
IFSR11	$\overline{\text{INT7}}$ Pin Polarity Select Bit (1)	0: One edge 1: Both edges	RW
IFSR12	$\overline{\text{INT8}}$ Pin Polarity Select Bit (1)	0: One edge 1: Both edges	RW
— (b5-b3)	No register bits; should be written with 0 and read as undefined value		—
IFSR16	UART5/UART6 Interrupt Source Select Bit	0: Bus collision, start condition detection, stop condition detection in UART5 1: Bus collision, start condition detection, stop condition detection in UART6	RW
— (b7)	No register bit; should be written with 0 and read as undefined value		—

Note:
1. This bit should be set to 0 to select the level sensitive input as trigger. To set this bit to 1, the POL bit in the corresponding INTiIC register (i = 6 to 8) should be set to 0 (falling edge).

Figure 17.19 IFSR1 Register

17.1 Synchronous Serial Interface Mode

The synchronous serial interface mode allows data transmission/reception synchronized with transmit/receive clock. Table 17.2 lists specifications of synchronous serial interface mode.

Table 17.2 Synchronous Serial Interface Mode Specifications

Item	Specification
Data format	8-bit character length
Transmit/receive clock	• The CKDIR bit in the UiMR register (i = 0 to 8) is set to 0 (internal clock): $\frac{fx}{2(m+1)}$ fx = f1, f8, f2n; m: UiBRG register setting value, 00h to FFh • The CKDIR bit is set to 1 (external clock): input into the CLKi pin
Transmit/receive control	Selectable among CTS, RTS, and CTS/RTS disabled
Transmit start conditions	The conditions for starting data transmission are as follows (1): • The TE bit in the UiC1 register is set to 1 (transmission enabled) • The TI bit in the UiC1 register is set to 0 (data held in the UiTB register) • Input level at the $\overline{\text{CTSi}}$ pin is L when the CTS is selected
Receive start conditions	The conditions for starting data reception are as follows (1): • The RE bit in the UiC1 register is set to 1 (reception enabled) • The TE bit in the UiC1 register is set to 1 (transmission enabled) • The TI bit in the UiC1 register is set to 0 (data held in the UiTB register) • Input level at the $\overline{\text{CTSi}}$ pin is low when the CTS is selected
Interrupt request generating timing	In transmit interrupt, one of the following conditions is selected to set the UiIRS bit in registers U0C1 to U6C1 and U78CON: • The UiIRS bit is set to 0 (transmit buffer in the UiTB register is empty): when data is transferred from the UiTB register to the UARTi transmit register (when the transmission has started) • The UiIRS bit is set to 1 (transmission is completed): when data transmission from the UARTi transmit register is completed In receive interrupt, • When data is transferred from the UARTi receive register to the UiRB register (when the reception is completed)
Error detection	Overrun error (2) This error occurs when the seventh bit of the next data has been received before the UiRB register is read
Selectable functions	• CLK polarity Selectable either rising or falling edge of the transmit/receive clock for output and input of transmit/receive data • Bit order selection Selectable either LSB first or MSB first • Continuous receive mode Data reception is enabled by a read access to the UiRB register • Serial data logical inversion (UART0 to UART6) This function logically inverses transmit/receive data

Notes:

1. In case external clock is selected, the following preconditions should be met:
   - The CLKi pin is held high when the CKPOL bit in the UiC0 register is set to 0 (transmit data output on the falling edge of the transmit/receive clock and receive data input on the rising edge)
   - The CLKi pin is held low when the CKPOL bit is set to 1 (transmit data output on the rising edge of the transmit/receive clock and receive data input on the falling edge)
2. If an overrun error occurs, the UiRB register is undefined. The IR bit in the SiRIC register is not changed to 1 (interrupts requested).

Table 17.3 and Table 17.4 list register settings. When UARTi (i = 0 to 8) operating mode is selected, a high is output at the TXDi pin until the transmission starts (the TXDi pin is high-impedance when the N-channel open drain output is selected).
Figure 17.20 and Figure 17.21 show respectively an example of transmit/receive operation in synchronous serial interface mode.

Table 17.3 Register Settings in Synchronous Serial Interface Mode (for UART0 to UART6)

Register	Bits	Function
UiMR	7 to 4	Set the bits to 0000b
	CKDIR	Select either the internal clock or the external clock
	SMD2 to SMD0	Set the bits to 001b
UiC0	UFORM	Select either LSB first or MSB first
	CKPOL	Select a transmit/receive clock polarity
	5	Set the bit to 0
	CRD	Select the CTS enabled or disabled
	TXEPT	Transmit register empty flag
	2	Set the bit to 0
	CLK1 and CLK0	Select a count source for the UiBRG register
UiC1	7	Set the bit to 0
	UiLCH	Set the bit to 1 to use logical inversion
	UiRRM	Set the bit to 1 to use continuous receive mode
	UiIRS	Select a source for UARTi transmit interrupt
	RI	Receive complete flag
	RE	Set the bit to 1 to enable data reception
	TI	Transmit buffer empty flag
	TE	Set the bit to 1 to enable data transmission/reception
UiSMR	7 to 0	Set the bits to 00h
UiSMR2	7 to 0	Set the bits to 00h
UiSMR3	7 to 0	Set the bits to 00h
UiSMR4	7 to 0	Set the bits to 00h
UiBRG	7 to 0	Set the bit rate
IFS0	IFS06	Select input pins for CLK3, RXD3, and $\overline{\text{CTS3}}$
	IFS03 and IFS02	Select input pins for CLK6, RXD6, and $\overline{\text{CTS6}}$
UiTB	7 to 0	Set the data to be transmitted
UiRB	OER	Overrun error flag
	7 to 0	Received data is read

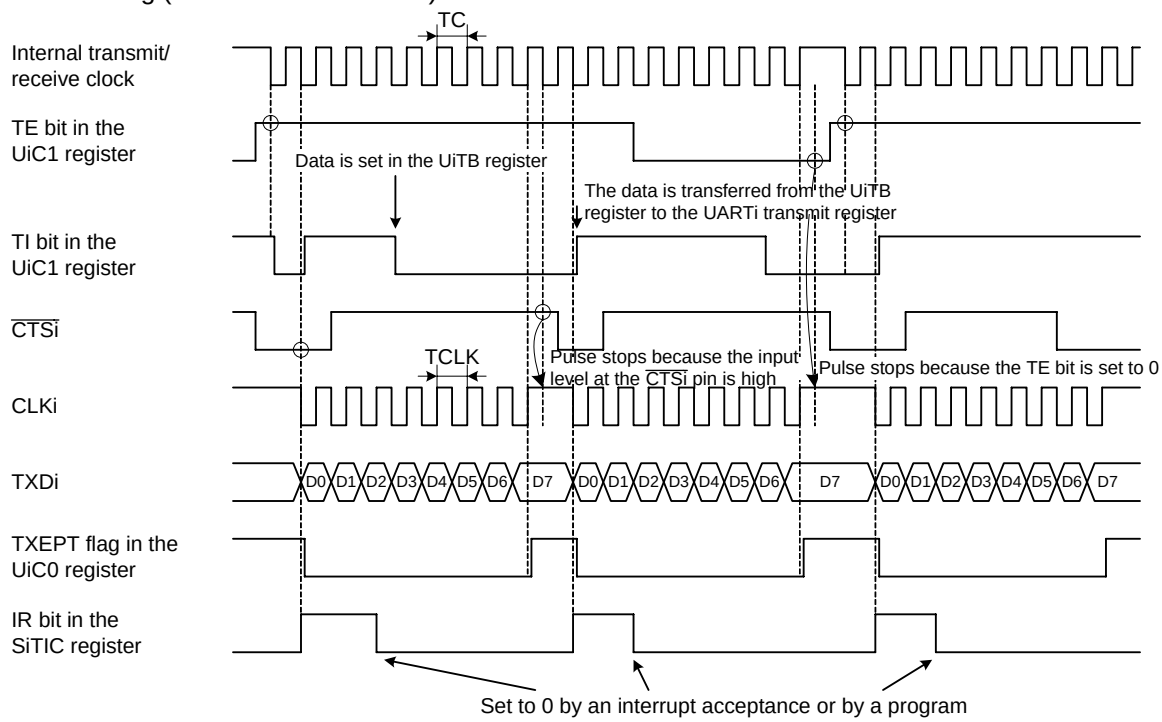
i = 0 to 6

Table 17.4 Register Settings in Synchronous Serial Interface Mode (for UART7 and UART8)

Register	Bits	Function
UiMR	7 to 4	Set the bits to 0000b
	CKDIR	Select the internal clock or the external clock
	SMD2 to SMD0	Set the bits to 001b
UiC0	UFORM	Select either LSB first or MSB first
	CKPOL	Select a transmit/receive clock polarity
	5	Set the bit to 0
	CRD	Select the CTS enabled or disabled
	TXEPT	Transmit register empty flag
	2	Set the bit to 0
	CLK1 and CLK0	Select a count source for the UiBRG register
UiC1	RI	Receive complete flag
	RE	Set the bit to 1 to enable data reception
	TI	Transmit buffer empty flag
	TE	Set the bit to 1 to enable data transmission/reception
U78CON	UiRRM	Set the bit to 1 to use continuous receive mode
	UiIRS	Select an interrupt source for UARTi transmit
IFS0	IFS05	Select input pins for CLK7, RXD7, and $\overline{\text{CTS7}}$
UiBRG	7 to 0	Set the bit rate
UiTB	7 to 0	Set the data to be transmitted
UiRB	OER	Overrun error flag
	7 to 0	Received data can be read

i = 7, 8

Transmit timing (in internal clock mode)

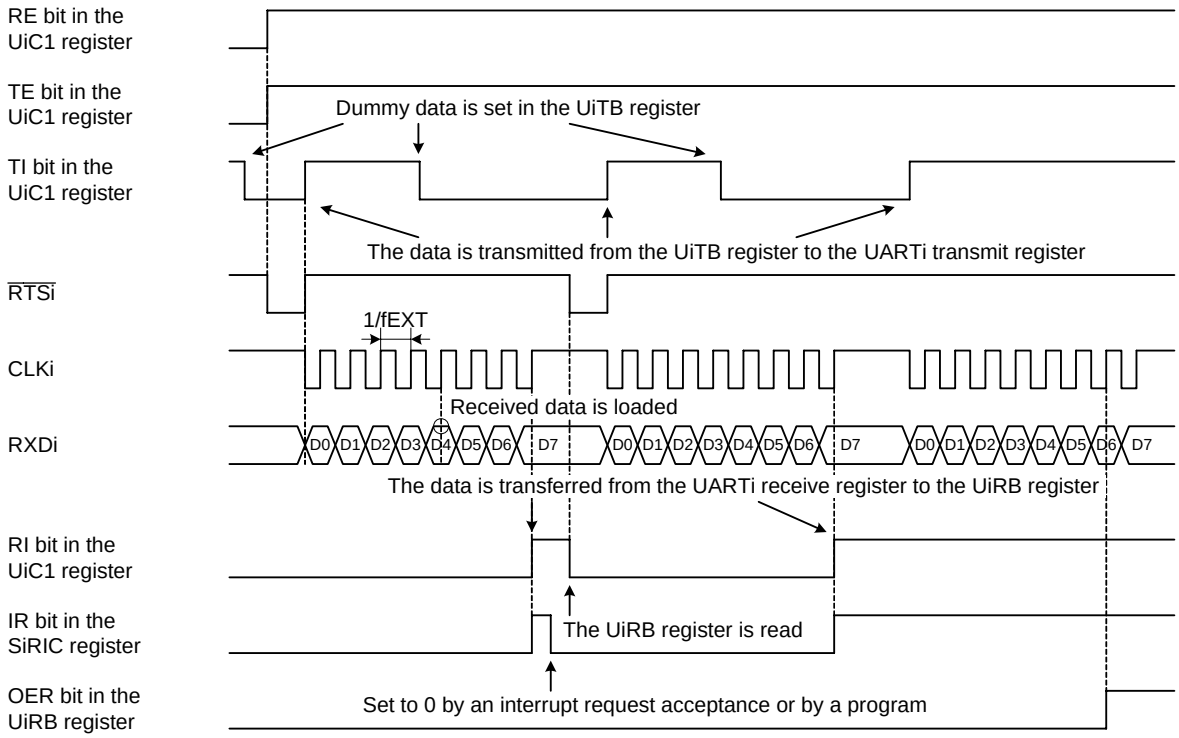

TC
Internal transmit/ receive clock
TE bit in the UiC1 register
Data is set in the UiTB register
The data is transferred from the UiTB register to the UARTi transmit register
TI bit in the UiC1 register
CTSi
TCLK
Pulse stops because the input level at the CTSi pin is high
Pulse stops because the TE bit is set to 0
CLKi
TXDi
D0 D1 D2 D3 D4 D5 D6 D7 D0 D1 D2 D3 D4 D5 D6 D7 D0 D1 D2 D3 D4 D5 D6 D7
TXEPT flag in the UiC0 register
IR bit in the SiTIC register
Set to 0 by an interrupt acceptance or by a program

The figure above applies under the following conditions:
- -The CKDIR bit in the UiMR register = 0 (internal clock selected)
- -The CRD bit in the UiC0 register = 0 (CTS enabled)
- -The CKPOL bit in the UiC0 register = 0 (transmitted data output on the falling edge of the transmit/receive clock)
- -The UiRS bit in registers UiC1 and U78CON = 0 (an transmit interrupt request generated because the UiTB register is empty)

TC = TCLK = 2(m+1)/fx
fx: UiBRG count source frequency (f1, f8, or f2n)
m: Setting value of the UiBRG register

Figure 17.20 Transmit Operation in Synchronous Serial Interface Mode

Receive timing (in external clock mode)


RE bit in the UiC1 register
TE bit in the UiC1 register
TI bit in the UiC1 register
RTSi
CLKi
RXDi
RI bit in the UiC1 register
IR bit in the SiRIC register
OER bit in the UiRB register
Dummy data is set in the UiTB register
The data is transmitted from the UiTB register to the UARTi transmit register
1/fEXT
Received data is loaded
D0 D1 D2 D3 D4 D5 D6 D7 D0 D1 D2 D3 D4 D5 D6 D7 D0 D1 D2 D3 D4 D5 D6 D7
The data is transferred from the UARTi receive register to the UiRB register
The UiRB register is read
Set to 0 by an interrupt request acceptance or by a program

The figure above applies under the following conditions:
- -The CKDIR bit in the UiMR register = 1 (external clock selected)
- -The CKPOL bit in the UiC0 register = 0 (received data input on the rising edge of the transmit/receive clock)

fEXT: External clock frequency

The following conditions should be met while an input level at the CLKi pin before receiving data is high:
- -The TE bit in the UiC1 register = 1 (transmission enabled)
- -The RE bit in the UiC1 register = 1 (reception enabled)
- -A write of dummy data to the UiTB register

Figure 17.21 Receive Operation in Synchronous Serial Interface Mode

17.1.1 Reset Procedure on Transmit/Receive Error

When a transmit/receive error occurs in synchronous serial interface mode, a reset is required as the procedure below:

A. Reset procedure for the UiRB register (i = 0 to 8)

(1) Set the RE bit in the UiC1 register to 0 (reception disabled).
(2) Set bits SMD2 to SMD0 in the UiMR register to 000b (serial interface disabled).
(3) Set bits SMD2 to SMD0 in the UiMR register to 001b (synchronous serial interface mode selected).
(4) Set the RE bit in the UiC1 register to 1 (reception enabled).

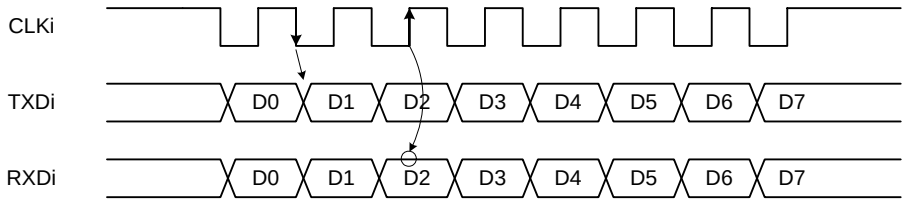
B. Reset procedure for the UiTB register

(1) Set bits SMD2 to SMD0 in the UiMR register to 000b (serial interface disabled).
(2) Set bits SMD2 to SMD0 in the UiMR register to 001b (synchronous serial interface mode selected).
(3) Set the TE bit in the UiC1 register to 1 (transmission enabled) irrespective of the bit setting.

17.1.2 CLK Polarity

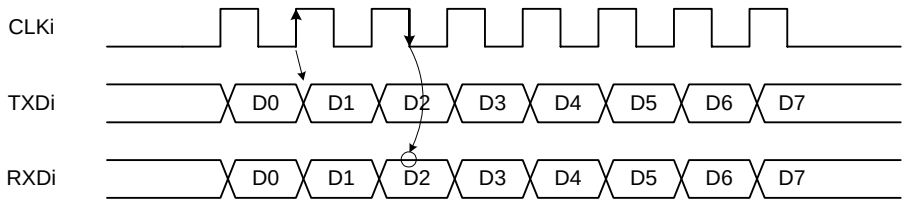
As shown in Figure 17.22, the polarity of the transmit/receive clock is selected using the CKPOL bit in the UiC0 register (i = 0 to 8).

Case 1: When the CKPOL bit in the UiC0 register is set to 0 (transmitted data output on the falling edge of the transmit/receive clock and received data input on the rising edge)



Notes:
1. The CLKi pin is held high when no data is transmitted/received.
2. The figure above applies under the following conditions:
-The UFORM bit in the UiC0 register = 0 (LSB first)
-The UiLCH bit in the UiC1 register = 0 (data is non logic-inverted)

Case 2: When the CKPOL bit in the UiC0 register is set to 1 (transmitted data output on the rising edge of the transmit/receive clock and received data input on the falling edge)



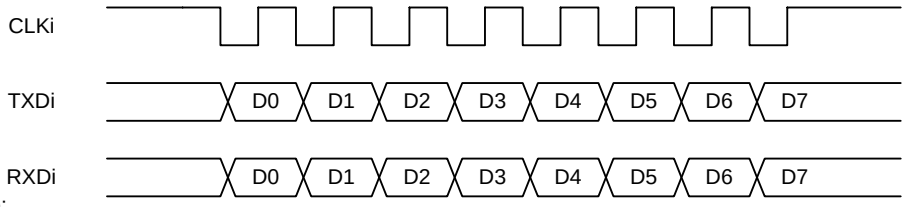
Notes:
3. The CLKi pin is held low when no data is transmitted/received.
4. The figure above applies under the following conditions:
-The UFORM bit in the UiC0 register = 0 (LSB first)
-The UiLCH bit in the UiC1 register = 0 (data is non logic-inverted)

Figure 17.22 Transmit/Receive Clock Polarity (i = 0 to 8)

17.1.3 LSB First and MSB First Selection

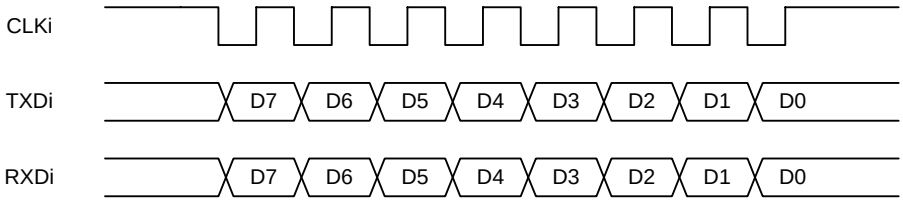
As shown in Figure 17.23, the bit order is selected using the UFORM bit in the UiC0 register (i = 0 to 8).

Case 1: When the UFORM bit in the UiC0 register is set to 0 (LSB first)



Note:
1. The figure above applies under the following conditions:
-The CKPOL bit in the UiC0 register = 0 (transmitted data output on the falling edge of the transmit/ receive clock and received data input on the rising edge)
-The UiLCH bit in the UiC1 register = 0 (data is non logic-inverted)

Case 2: When the UFORM bit in the UiC0 register is set to 1 (MSB first)



Note:
2. The figure above applies under the following conditions:
-The CKPOL bit in the UiC0 register = 0 (transmitted data output on the falling edge of the transmit/ receive clock and received data input on the rising edge)
-The UiLCH bit in the UiC1 register = 0 (data is non logic-inverted)

Figure 17.23 Bit Order (i = 0 to 8)

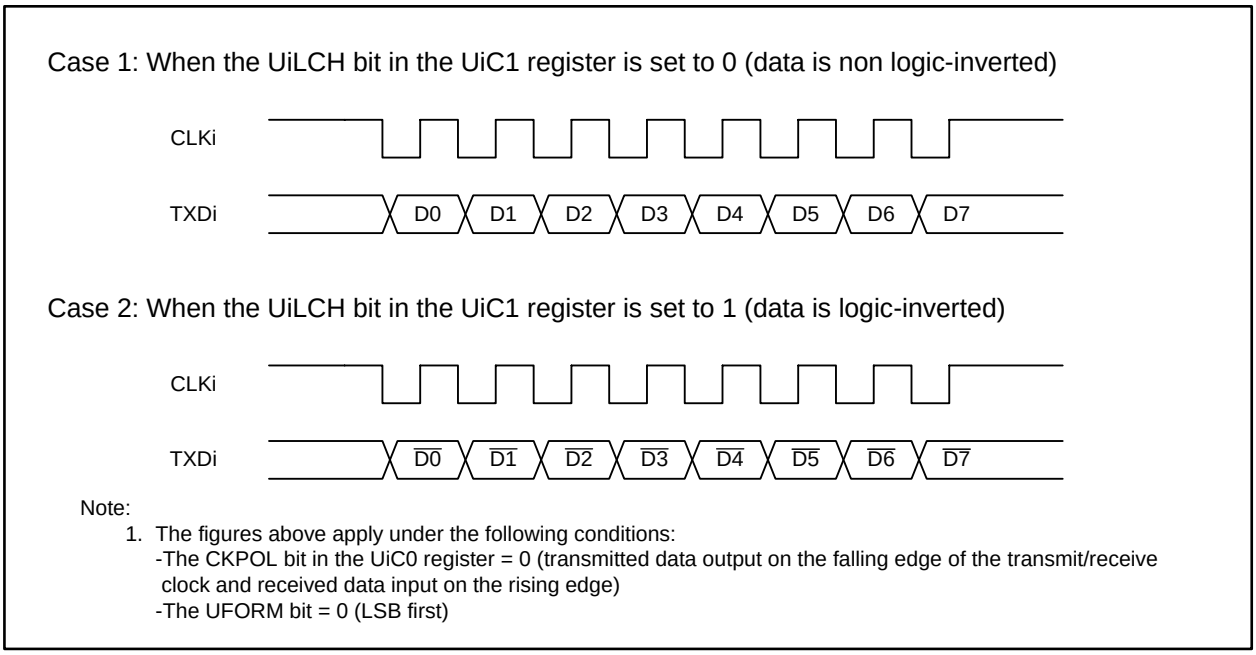
17.1.4 Continuous Receive Mode

In continuous receive mode, data reception is automatically enabled by a read access to the receive buffer register without any write of dummy data to the transmit buffer register. To start data reception, however, dummy data is required to read the receive buffer register.

When the UiRRM bit (i = 0 to 8) in registers U0C1 to U6C1 and U78CON is set to 1 (continuous receive mode enabled), the TI bit in the UiC1 register is set to 0 (data held in the UiTB register) by a read access to the UiRB register. In this UiRRM bit setting, any dummy data should not be written to the UiTB register.

17.1.5 Serial Data Logical Inversion

When the UiLCH bit in the UiC1 register (i = 0 to 6) is set to 1 (data logic-inverted), logical value written in the UiTB register is inverted to be transmitted. The UiRB register is read as logic-inverted receive data. Figure 17.24 shows the logical inversion of serial data.



Note:
1. The figures above apply under the following conditions:
-The CKPOL bit in the UiC0 register = 0 (transmitted data output on the falling edge of the transmit/receive clock and received data input on the rising edge)
-The UFORM bit = 0 (LSB first)

Figure 17.24 Serial Data Logical Inversion (i = 0 to 6)

17.1.6 CTS/RTS Function

The CTS controls data transmission using the $\overline{\text{CTSi}}/\overline{\text{RTSi}}$ pin (i = 0 to 8). When an input level at the pin becomes low, data transmission is started. If the input level changes to high during transmit operation, the transmission of the next data is stopped.

In synchronous serial interface mode, the transmitter is required to operate even during the receive operation. If the CTS is enabled, the input level at the $\overline{\text{CTSi}}/\overline{\text{RTSi}}$ pin should be low to start data reception as well.

The RTS indicates receiver status using the $\overline{\text{CTSi}}/\overline{\text{RTSi}}$ pin. When data reception is ready, the output level at the pin becomes low. It becomes high on the first falling edge of the CLKi pin.

17.2 Asynchronous Serial Interface Mode (UART Mode)

The UART mode enables data transmission/reception synchronized with an internal clock generated by a trigger on the falling edge of the start bit. Table 17.5 lists specifications of UART mode.

Table 17.5 UART Mode Specifications

Item	Specification
Data format	• Start bit: 1 bit-length • Data bit (data character) selectable among 7, 8, and 9 bit-length • Parity bit: selectable among odd, even, and none • Stop bit: selectable between 1 and 2 bit-length
Transmit/receive clock	• The CKDIR bit in the UiMR register (i = 0 to 8) is set to 0 (internal clock): $\frac{fx}{16(m+1)}$ fx = f1, f8, f2n; m: UiBRG register setting value, 00h to FFh • The CKDIR bit is set to 1 (external clock) $\frac{fEXT}{16(m+1)}$ fEXT: Clock applied to the CLKi pin
Transmit/receive control	Selectable among CTS, RTS, and CTS/RTS disabled
Transmit start conditions	The conditions for starting data transmission are as follows: • The TE bit in the UiC1 register is set to 1 (transmission enabled) • The TI bit in the UiC1 register is set to 0 (data held in the UiTB register) • Input level at the $\overline{\text{CTSi}}$ pin is low when the CTS is selected
Receive start conditions	The conditions for starting data reception are as follows: • The RE bit in the UiC1 register is set to 1 (reception enabled) • The start bit is detected
Interrupt request generating timing	In transmit interrupt, one of the following conditions is selected to set the UiIRS bit in registers U0C1 to U6C1 and U78CON: • The UiIRS bit is set to 0 (transmit buffer in the UiTB register is empty): when data is transmitted from the UiTB register to the UARTi transmit register (when the transmission has started) • The UiIRS bit is set to 1 (transmission is completed): when data transmission from the UARTi transmit register is completed In receive interrupt, • When data is transmitted from the UARTi receive register to the UiRB register (when reception is completed)
Error detection	• Overrun error (1) This error occurs when one bit prior to the stop bit (when 1 stop bit length is selected) or the first stop bit (when 2 stop bit length is selected) of the next data has been received before the UiRB register is read • Framing error This error occurs when the required number of stop bits is not detected • Parity error This error occurs when an even number of 1's in parity and character bits is detected while the odd number is set, or vice versa. The parity should be enabled • Error sum flag This flag is set to 1 when any of overrun error, framing error, or parity error occurs
Selectable functions	• Bit order selection Selectable either LSB first or MSB first • Serial data logical inversion This function logically inverses transmit/receive data. The start bit and stop bit are not inverted • TXD/RXD I/O polarity switching The output level from the TXD pin and the input level to the RXD pin are inverted. All I/O levels are inverted

Note:

1. When an overrun error occurs, the UiRB register is undefined. The IR bit in the SiRIC register does not change to 1 (interrupts requested).

Table 17.6 and Table 17.7 list register settings. When UARTi (i = 0 to 8) operating mode is selected, a high is output at the TXDi pin until the transmission starts (the TXDi pin is high-impedance when the N-channel open drain output is selected). Figure 17.25 and Figure 17.26 show examples of transmit operation in UART mode. Figure 17.27 shows an example of receive operation.

Table 17.6 Register Settings in UART Mode (UART0 to UART6)

Register	Bits	Function
UiMR	IOPOL	Select I/O polarity of pins TXD and RXD
	PRY and PRYE	Select parity enabled or disabled, and odd or even
	STPS	Select a stop bit length
	CKDIR	Select the internal clock or the external clock
	SMD2 to SMD0	Set the bits to 100b in 7-bit character length
		Set the bits to 101b in 8-bit character length
		Set the bits to 110b in 9-bit character length
UiC0	UFORM	Selectable either LSB first or MSB first in 8-bit character length. Set the bit to 0 in 7-bit or 9-bit character length
	CKPOL	Set the bit to 0
	5	Set the bit to 0
	CRD	Select the CTS enabled or disabled
	TXEPT	Transmit register empty flag
	2	Set the bit to 0
	CLK1 and CLK0	Select a count source for the UiBRG register
UiC1	7	Set the bit to 0
	UiLCH	Set the bit to 1 to use logical inversion
	UiRRM	Set the bit to 0
	UiIRS	Select an interrupt source for UARTi transmit
	RI	Receive complete flag
	RE	Set the bit to 1 to enable data reception
	TI	Transmit buffer empty flag
	TE	Set the bit to 1 to enable data transmission
UiSMR	7 to 0	Set the bits to 00h
UiSMR2	7 to 0	Set the bits to 00h
UiSMR3	7 to 0	Set the bits to 00h
UiSMR4	7 to 0	Set the bits to 00h
UiBRG	7 to 0	Set the bit rate
IFS0	IFS06	Select input pins for CLK3, RXD3, and $\overline{\text{CTS3}}$
	IFS03 and IFS02	Select input pins for CLK6, RXD6, and $\overline{\text{CTS6}}$
UiTB	8 to 0	Set the data to be transmitted (1)
UiRB	OER, FER, PER, and SUM	Error flag
	8 to 0	Received data is read (1)

i = 0 to 6

Note:
1. The bits to be used are as follows: 7-bit character length: bits 6 to 0
   8-bit character length: bits 7 to 0
   9-bit character length: bits 8 to 0

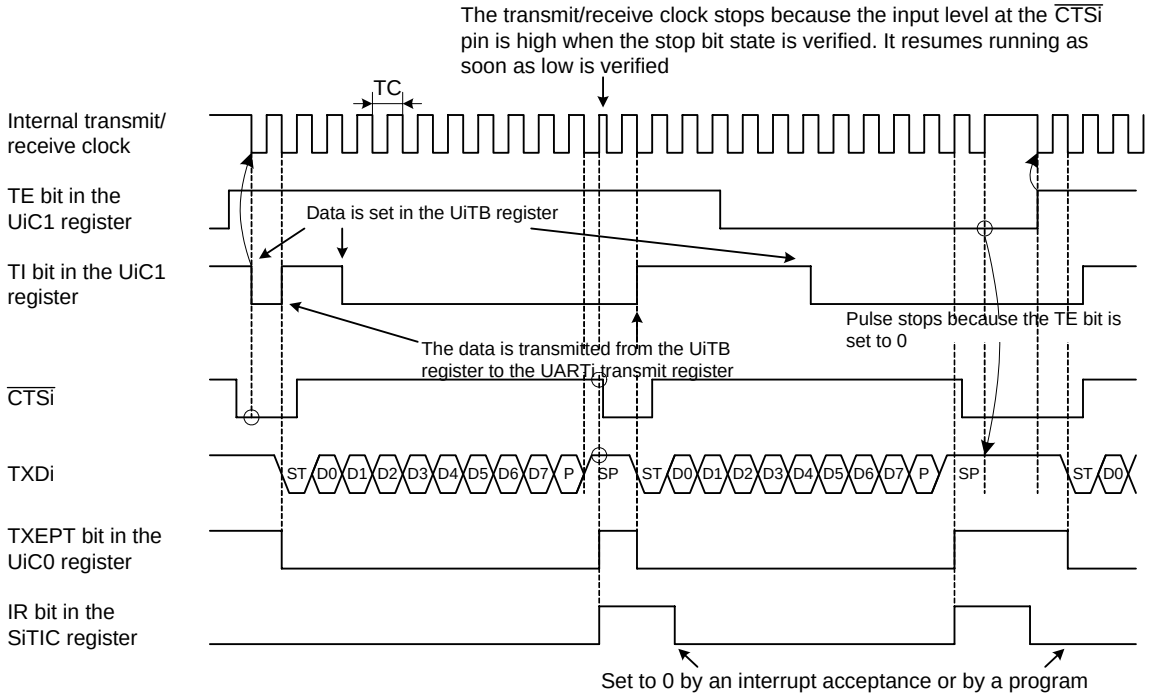
Table 17.7 Register Settings in UART Mode (UART7, UART8)

Register	Bits	Function
UiMR	PRY and PRYE	Select parity enabled or disabled, and odd or even
	STPS	Select a stop bit length
	CKDIR	Select the internal clock or the external clock
	SMD2 to SMD0	Set the bits to 100b in 7-bit character length
		Set the bits to 101b in 8-bit character length
		Set the bits to 110b in 9-bit character length
UiC0	UFORM	Selectable either LSB first or MSB first in 8-bit character length. Set the bit to 0 in 7-bit or 9-bit character length
	CKPOL	Set the bit to 0
	5	Set the bit to 0
	CRD	Select the CTS enabled or disabled
	TXEPT	Transmit register empty flag
	2	Set the bit to 0
	CLK1 and CLK0	Select a count source for the UiBRG register
UiC1	RI	Receive complete flag
	RE	Set the bit to 1 to enable data reception
	TI	Transmit buffer empty flag
	TE	Set the bit to 1 to enable data transmission
U78CON	UiRRM	Set the bit to 0
	UiIRS	Select an interrupt source for UARTi transmit
UiBRG	7 to 0	Set the bit rate
IFS0	IFS05	Select input pins for CLK7, RXD7, and $\overline{\text{CTS7}}$
UiTB	8 to 0	Set the data to be transmitted (1)
UiRB	OER, FER, PER, and SUM	Error flag
	8 to 0	Received data is read (1)

i = 7, 8

Note:
1. The bits to be used are as follows: 7-bit character length: bits 6 to 0
   8-bit character length: bits 7 to 0
   9-bit character length: bits 8 to 0

8-bit character long data transmit timing (parity enabled, 1 stop bit)


The transmit/receive clock stops because the input level at the CTSi pin is high when the stop bit state is verified. It resumes running as soon as low is verified
TC
Internal transmit/ receive clock
TE bit in the UiC1 register
Data is set in the UiTB register
TI bit in the UiC1 register
Pulse stops because the TE bit is set to 0
The data is transmitted from the UiTB register to the UARTi transmit register
CTSi
TXDi
ST D0 D1 D2 D3 D4 D5 D6 D7 P SP ST D0 D1 D2 D3 D4 D5 D6 D7 P SP ST D0
TXEPT bit in the UiC0 register
IR bit in the SiTIC register
Set to 0 by an interrupt acceptance or by a program

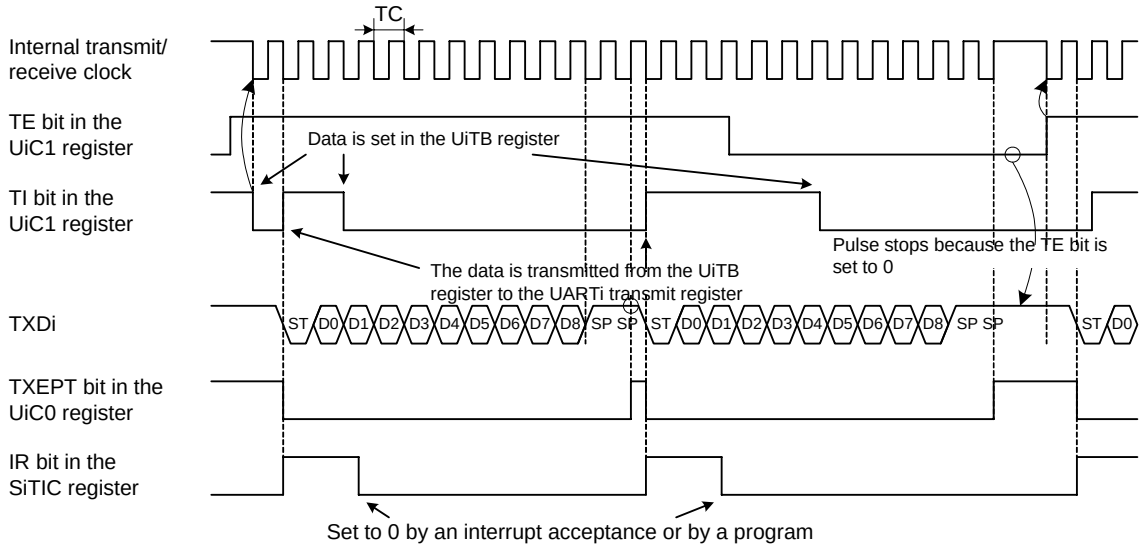
ST: Start bit
P: Parity bit
SP: Stop bit

The figure above applies under the following conditions:
- -The PRYE bit in the UiMR register = 1 (parity enabled)
- -The STPS bit in the UiMR register = 0 (1 stop bit)
- -The CRD bit in the UiC0 register = 0 (CTS enabled)
- -The UiIRS bit in the UiC1 register = 1 (transmit interrupt request generated because the UiTB register is empty)

TC = 16(m+1)/fx or 16(m+1)/fEXT
fx: UiBRG count source frequency (f1, f8, f2n)
fEXT: UiBRG count source frequency (external clock)
m: Setting value of the UiBRG register

Figure 17.25 Transmit Operation in UART Mode (1) (i = 0 to 8)

9-bit character long data transmit timing (parity disabled, 2 stop bits)


TC
Internal transmit/ receive clock
TE bit in the UiC1 register
Data is set in the UiTB register
TI bit in the UiC1 register
The data is transmitted from the UiTB register to the UARTi transmit register
Pulse stops because the TE bit is set to 0
TXDi
ST D0 D1 D2 D3 D4 D5 D6 D7 D8 SP SP ST D0 D1 D2 D3 D4 D5 D6 D7 D8 SP SP ST D0
TXEPT bit in the UiC0 register
IR bit in the SiTIC register
Set to 0 by an interrupt acceptance or by a program

ST: Start bit
P: Parity bit
SP: Stop bit

The figure above applies under the following conditions:
- -The PRYE bit in the UiMR register = 0 (parity disabled)
- -The STPS bit in the UiMR register = 1 (2 stop bits)
- -The CRD bit in the UiC0 register = 1 (CTS disabled)
- -The UiIRS bit in the UiC1 register = 0 (transmit interrupt request generated because the transmission is completed)

TC = 16(m+1)/fx or 16(m+1)/fEXT
fx: UiBRG count source frequency (f1, f8, f2n)
fEXT: UiBRG count source frequency (external clock)
m: Setting value of the UiBRG register

Figure 17.26 Transmit Operation in UART Mode (2) (i = 0 to 8)

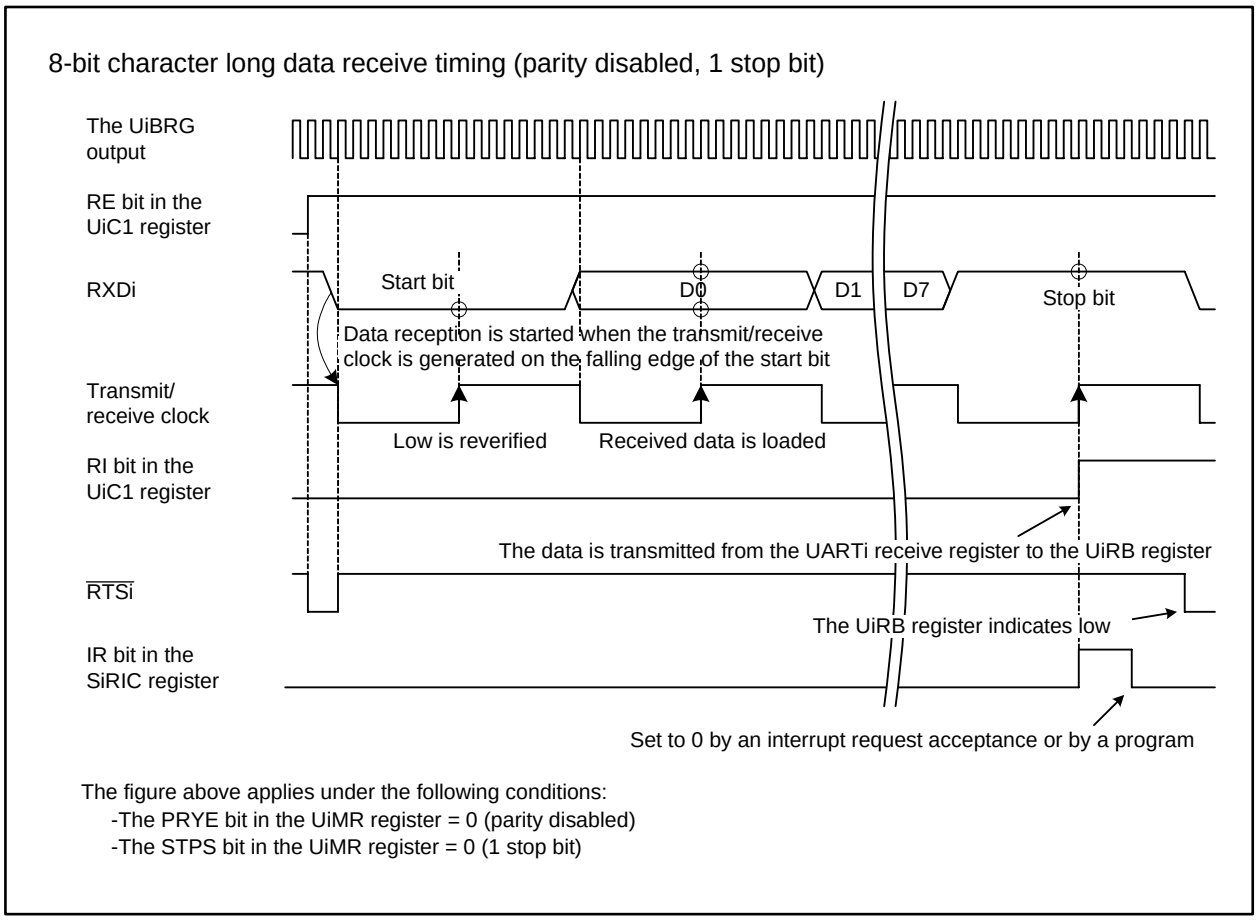

8-bit character long data receive timing (parity disabled, 1 stop bit)
The UiBRG output
RE bit in the UiC1 register
RXDi
Start bit
D0
D1
D7
Stop bit
Data reception is started when the transmit/receive clock is generated on the falling edge of the start bit
Transmit/ receive clock
Low is reverified
Received data is loaded
RI bit in the UiC1 register
The data is transmitted from the UARTi receive register to the UiRB register
RTSi
The UiRB register indicates low
IR bit in the SiRIC register
Set to 0 by an interrupt request acceptance or by a program
The figure above applies under the following conditions:
-The PRYE bit in the UiMR register = 0 (parity disabled)
-The STPS bit in the UiMR register = 0 (1 stop bit)

Figure 17.27 Receive Operation in UART mode (i = 0 to 8)

17.2.1 Bit Rate

In UART mode, the bit rate is clock frequency which is divided by a setting value of the UiBRG register (i = 0 to 8) and again divided by 16. Table 17.8 lists an example of bit rate setting.

Table 17.8 Bit Rate Setting

Bit Rate (bps)	Count Source of BRG	Peripheral Clock: 30 MHz		Peripheral Clock: 32 MHz	
		Setting value of BRG: n	Actual bit rate (bps)	Setting value of BRG: n	Actual bit rate (bps)
1200	f8	194 (C2h)	1202	207 (CHh)	1202
2400	f8	97 (61h)	2392	103 (67h)	2404
4800	f8	48 (30h)	4783	51 (33h)	4808
9600	f1	194 (C2h)	9615	207 (CFh)	9615
14400	f1	129 (81h)	14423	138 (8Ah)	14388
19200	f1	97 (61h)	19133	103 (67h)	19231
28800	f1	64 (40h)	28846	68 (44h)	28986
31250	f1	59 (3Bh)	31250	63 (3Fh)	31250
38400	f1	48 (30h)	38265	51 (33h)	38462
51200	f1	36 (24h)	50676	38 (26h)	51282

17.2.2 Reset Procedure on Transmit/Receive Error

When a transmit/receive error occurs in UART mode, a reset is required as the procedure below:

A. Reset procedure for the UiRB register (i = 0 to 8)

(1) Set the RE bit in the UiC1 register to 0 (reception disabled).
(2) Set the RE bit in the UiC1 register to 1 (reception enabled).

B. Reset procedure for the UiTB register

(1) Set bits SMD2 to SMD0 in the UiMR register to 000b (serial interface disabled).
(2) Set again bits SMD2 to SMD0 to either of 001b, 101b, or 110b.
(3) Set the TE bit in the UiC1 register to 1 (transmission enabled) irrespective of the bit setting.

17.2.3 LSB First and MSB First Selection

As shown in Figure 17.28, the bit order is selected using the UFORM bit in the UiC0 register (i = 0 to 8). This function is available for the data format of 8-bit character length.

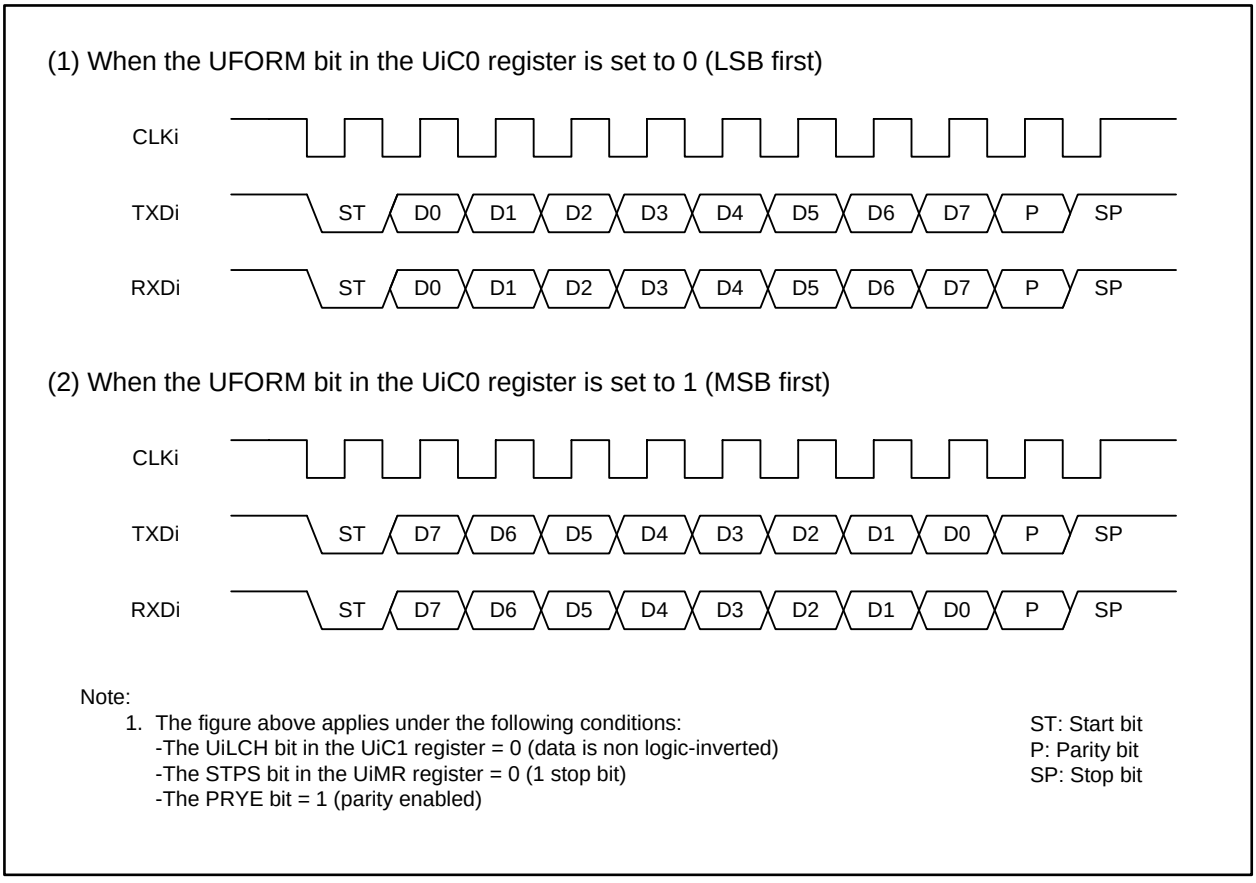

(1) When the UFORM bit in the UiC0 register is set to 0 (LSB first)
CLKi
TXDi
RXDi
ST D0 D1 D2 D3 D4 D5 D6 D7 P SP
(2) When the UFORM bit in the UiC0 register is set to 1 (MSB first)
CLKi
TXDi
RXDi
ST D7 D6 D5 D4 D3 D2 D1 D0 P SP
Note:
1. The figure above applies under the following conditions:
-The UiLCH bit in the UiC1 register = 0 (data is non logic-inverted)
-The STPS bit in the UiMR register = 0 (1 stop bit)
-The PRYE bit = 1 (parity enabled)
ST: Start bit
P: Parity bit
SP: Stop bit

Figure 17.28 Bit Order (i = 0 to 8)

17.2.4 Serial Data Logical Inversion

When the UiLCH bit in the UiC1 register (i = 0 to 6) is set to 1 (data logic-inverted), logical value written in the UiTB register is inverted to be transmitted. The UiRB register is read as logic-inverted receive data. The parity bit is not inverted. Figure 17.29 shows the logical inversion of serial data.

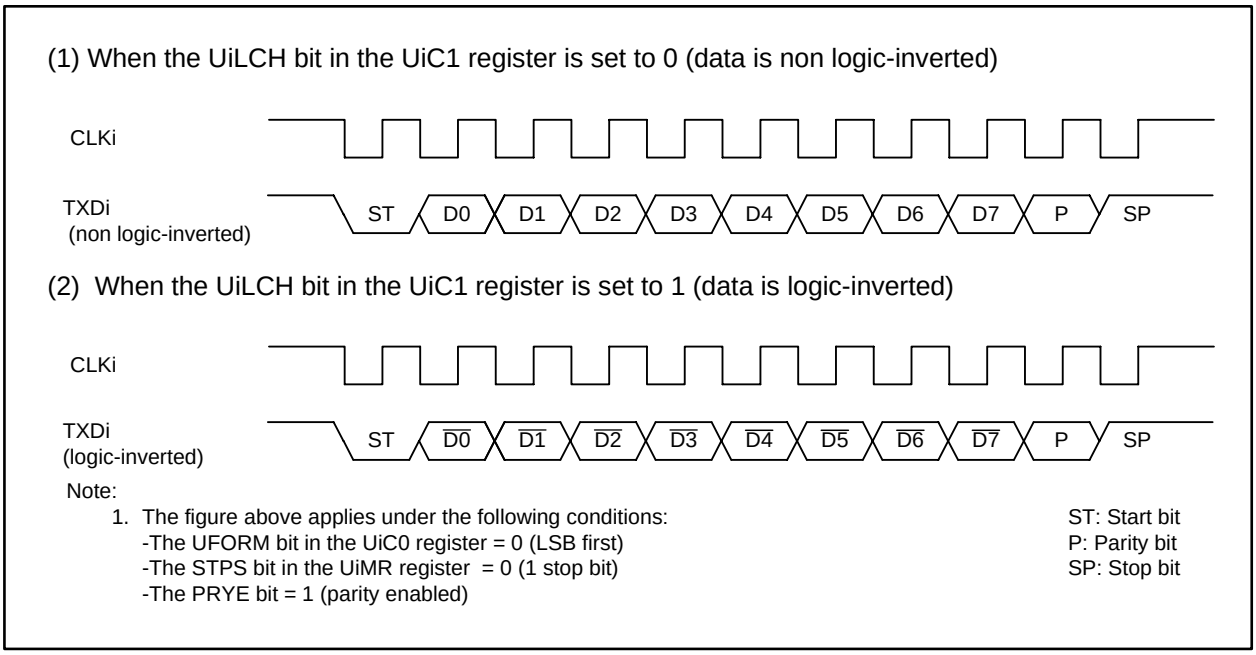


Figure 17.29 Serial Data Logical Inversion (i = 0 to 6)

17.2.5 TXD and RXD I/O Polarity Inversion

The output level at the TXD pin and the input level at the RXD pin are inverted by this function. All I/O data levels, including the start bit, stop bit, and parity bit are inverted by setting the IOPOL bit in the UiMR register (i = 0 to 6) to 1 (inverted). Figure 17.30 shows TXD and RXD I/O polarity inversion.

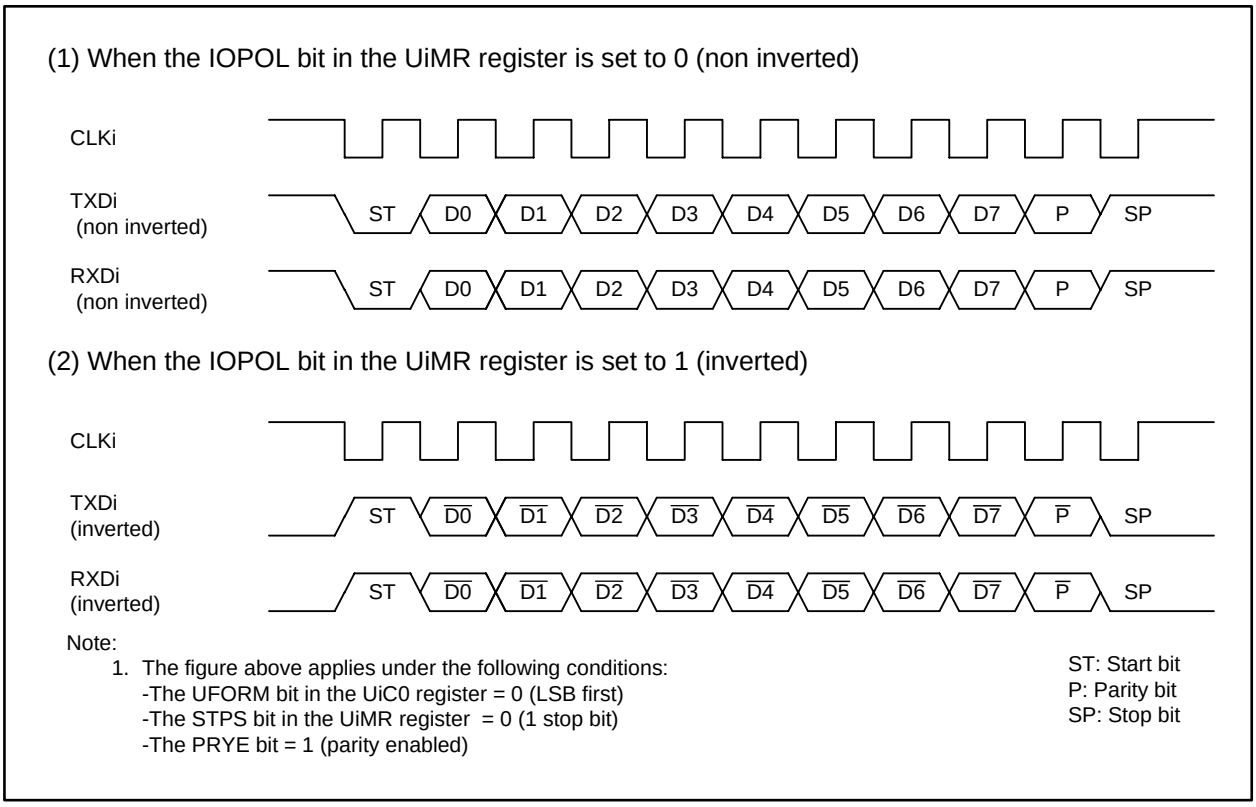

(1) When the IOPOL bit in the UiMR register is set to 0 (non inverted)
CLKi
TXDi (non inverted)
RXDi (non inverted)
ST D0 D1 D2 D3 D4 D5 D6 D7 P SP
(2) When the IOPOL bit in the UiMR register is set to 1 (inverted)
CLKi
TXDi (inverted)
RXDi (inverted)
ST D0 D1 D2 D3 D4 D5 D6 D7 P SP
Note:
1. The figure above applies under the following conditions:
-The UFORM bit in the UiC0 register = 0 (LSB first)
-The STPS bit in the UiMR register = 0 (1 stop bit)
-The PRYE bit = 1 (parity enabled)
ST: Start bit
P: Parity bit
SP: Stop bit

Figure 17.30 TXD and RXD I/O Polarity Inversion (i = 0 to 6)

17.2.6 CTS/RTS Function

The CTS controls data transmission using the $\overline{\text{CTSi}}/\overline{\text{RTSi}}$ pin (i = 0 to 8). When an input level at the pin becomes low, data transmission is started. If the input level changes to high during transmit operation, the transmission of the next data is stopped.

The RTS indicates receiver status using the $\overline{\text{CTSi}}/\overline{\text{RTSi}}$ pin. When the MCU is ready to receive data, the output level at the pin becomes low. It becomes high on the first falling edge of the CLKi pin.

17.3 Special Mode 1 (I^2C Mode)

This mode uses an I^2C-typed interface for communication. Table 17.9 lists specifications of the I^2C mode.

Table 17.9 I^2C Mode Specifications

Item	Specification
Data format	8-bit character length
Transmit/receive clock	In master mode • The CKDIR bit in the UiMR register (i = 0 to 6) is set to 0 (internal clock): $\frac{fx}{2(m+1)}$ fx = f1, f8, f2n m: UiBRG register setting value, 00h to FFh In slave mode • The CKDIR bit is set to 1 (external clock): input into the SCLi pin
Transmit start conditions	The conditions for starting data transmission are as follows (1): • The TE bit in the UiC1 register is set to 1 (transmission enabled) • The TI bit in the UiC1 register is set to 0 (data held in the UiTB register)
Receive start conditions	The conditions for starting data reception are as follows (1): • The RE bit in the UiC1 register is set to 1 (reception enabled) • The TE bit in the UiC1 register is set to 1 (transmission enabled) • The TI bit in the UiC1 register is set to 0 (data held in the UiTB register)
Interrupt request generating timing	When any of the following is detected: start condition, stop condition, NACK (Not-acknowledge), or ACK (Acknowledge)
Error detection	Overrun error (2) This error occurs when the eighth bit of the next data has been received before the UiRB register is read
Selectable functions	• Arbitration lost Selectable update timing of the ABT bit in the UiRB register • SDAi digital delay Selectable among no digital delay and two to eight cycles of digital delay of UiBRG count source • Clock phase setting Selectable either clock delayed or no clock delay

Notes:
1. When an external clock is selected, the conditions should be met while the external clock signal is held high.
2. If an overrun error occurs, the UiRB register is undefined. The IR bit in the SiRIC register is not changed to 1 (interrupts requested).

Table 17.10 and Table 17.11 list respectively register settings and functions in I^2C mode. Figure 17.31 shows a block diagram of I^2C mode and Figure 17.32 shows timings for the transfer to the UiRB register (i = 0 to 6) and the interrupt.

As shown in Table 17.11, this mode is available when bits SMD2 to SMD0 in the UiMR register (i = 0 to 6) are set to 010b, and the IICM bit in the UiSMR register is set to 1. Since a transmit signal at the SDAi pin is output via the delay circuit, it changes after the SCLi pin is stably held low.

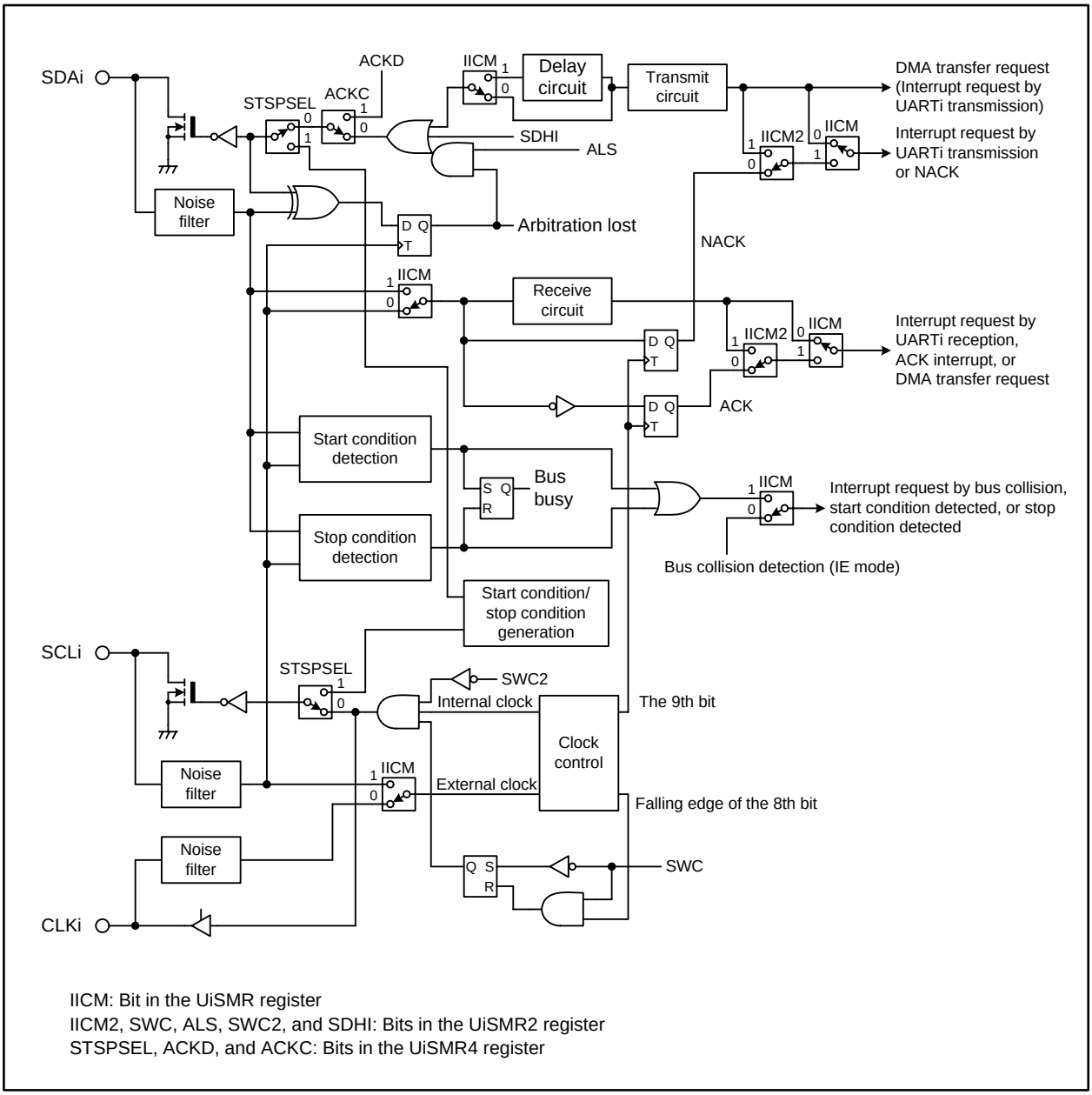


Figure 17.31 I²C Mode Block Diagram (i = 0 to 6)

Table 17.10 Register Settings in I²C Mode (i = 0 to 6)

Register	Bits	Function	
		Master	Slave
UiMR	IOPOL	Set the bit to 0	
	CKDIR	Set the bit to 0	Set the bit to 1
	SMD2 to SMD0	Set the bit to 010b	
UiC0	7 to 4	Set the bits to 1001b	
	TXEPT	Transmit register empty flag	
	2	Set the bit to 0	
	CLK1 and CLK0	Select a count source for the UiBRG register	Disabled
UiC1	7 to 5	Set the bits to 000b	
	UiIRS	Set the bit to 1	
	RI	Receive complete flag	
	RE	Set the bit to 1 to enable data reception	
	TI	Transmit buffer empty flag	
	TE	Set the bit to 1 to enable data transmission/reception	
UiSMR	7 to 3	Set the bits to 00000b	
	BBS	Bus busy flag	
	ABC	Select an arbitration lost detection timing	Disabled
	IICM	Set the bit to 1	
UiSMR2	7	Set the bit to 0	
	SDHI	Set the bit to 1 to disable the SDA output	
	SWC2	Set the bit to 1 to hold the SCL output at a forcible low	
	STC	Set the bit to 0	Set the bit to 1 to reset UARTi by detecting the start condition
	ALS	Set the bit to 1 to stop the output at the SDAi pin to detect an arbitration lost	Set the bit to 0
	SWC	Set the bit to 1 to hold a low output at the SCLi pin after receiving the eighth bit of the clock	
	CSC	Set the bit to 1 to enable clock synchronization	Set the bit to 0
	IICM2	Refer to Table 17.11	
UiSMR3	DL2 to DL0	Set the digital delay value of SDAi	
	4 to 2	Set the bit to 000b	
	CKPH	Refer to Table 17.11	
	SSE	Set the bit to 0	
UiSMR4	SWC9	Set the bit to 0	Set the bit to 1 to hold a low output at the SCLi pin after receiving the ninth bit of the clock
	SCLHI	Set the bit to 1 to stop the SCL output to detect stop condition	Set the bit to 0
	ACKC	Set the bit to 1 for ACK data output	
	ACKD	Select ACK or NACK	
	STSPSEL	Set the bit to 1 when any condition is output	Set the bit to 0
	STPREQ	Set the bit to 1 to generate a stop condition	Set the bit to 0
	RSTAREQ	Set the bit to 1 to generate a restart condition	Set the bit to 0
	STAREQ	Set the bit to 1 to generate a start condition	Set the bit to 0
UiBRG	7 to 0	Set the bit rate	Disabled
IFSR0	IFSR06 and IFSR07	Select a UART as interrupt source	
IFSR1	IFSR16	Select a UART as interrupt source	
IFS0	IFS06	Select input pins for SCL3 and SDA3	
	IFS03 and IFS02	Select input pins for SCL6 and SDA6	
UiTB	8	Set the bit to 1 on transmission. Set the bit to the value of the ACK bit on reception	
	7 to 0	Set the data to be transmitted on transmission. Set the register to FFh on reception	
UiRB	OER	Overrun error flag	
	ABT	Arbitration lost detection flag	Disabled
	8	D0 is loaded immediately after a receive interrupt is generated. ACK or NACK is loaded after a transmit interrupt is generated	
	7 to 0	D7 to D1 are read immediately after a receive interrupt is generated. D7 to D0 are read after a transmit interrupt is generated	

Table 17.11 I²C Mode Functions (i = 0 to 6)

Function	Synchronous Serial Interface Mode (SMD2 to SMD0 = 001b, IICM = 0)	I²C Mode (SMD2 to SMD0 = 010b, IICM = 1)			
		IICM2 = 0 (ACK/NACK interrupt)		IICM2 = 1 (Transmit/receive interrupt)	
		CKPH = 0 (No clock delay)	CKPH = 1 (Clock delayed)	CKPH = 0 (No clock delay)	CKPH = 1 (Clock delayed)
Source of software interrupt numbers 6, 39 to 41 (1) (Refer to Figure 17.32)	—	Start condition or stop condition detection (Refer to Table 17.12)			
Source of software interrupt numbers 2, 4, 17, 19, 33, 35, and 37 (1) (Refer to Figure 17.32)	UARTi transmission: Transmission started or completed (selected using the UiIRS register)	NACK detection: Rising edge of the ninth bit of SCLi		UARTi transmission: Rising edge of the ninth bit of SCLi	UARTi transmission: Falling edge of the ninth bit of SCLi
Source of software interrupt numbers 3, 5, 18, 20, 34, 36, and 38 (1) (Refer to Figure 17.32)	UARTi reception: Receiving at eighth bit CKPOL = 0 (rising edge) CKPOL = 1 (falling edge)	ACK detection: Rising edge of the ninth bit of SCLi		UARTi reception: Falling edge of the eighth bit of SCLi	
Data transfer timing from the UART receive register to the UiRB register	CKPOL = 0 (rising edge) CKPOL = 1 (falling edge)	Rising edge of the ninth bit of SCLi		Falling edge of the eighth bit of SCLi	Falling edge of the eighth bit and rising edge of the ninth bit of SCLi
UARTi transmit output delay	No delay	Delayed			
Pins P6_3, P6_7, P7_0, P7_3, P7_6, P9_2, P9_6, P11_0, P12_0, P15_0, and P15_4	TXDi output	SDAi I/O			
Pins P6_2, P6_6, P7_1, P7_5, P8_0, P9_1, P9_7, P11_2, P12_2, P15_2, and P15_5	RXDi input	SCLi I/O			
Pins P6_1, P6_5, P7_2, P7_4, P7_7, P9_0, P9_5, P11_1, P12_1, P15_1, and P15_6	Select CLKi input or output	— (Not used in I²C mode)			
Read level at pins RXDi and SCLi	Readable irrespective of the port direction bit				
Default output value at the SDAi pin	—	High (Value set in the Port Pi register (i = 0 to 7) if the I/O port is selected by output function select registers)			
SCLi default and end values	—	High	Low	High	Low
DMA source (Refer to Figure 17.32)	UARTi reception	ACK detection		UARTi reception: Falling edge of the eighth bit of SCLi	

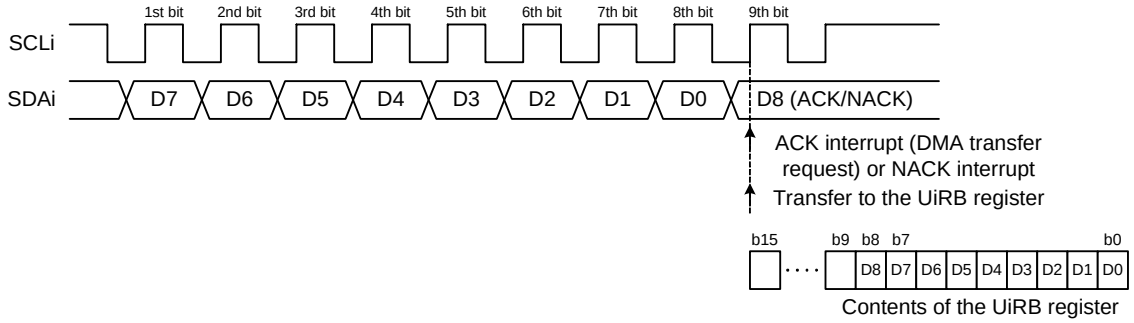
Table 17.11 I²C Mode Functions (i = 0 to 6)

Function	Synchronous Serial Interface Mode (SMD2 to SMD0 = 001b, IICM = 0)	I²C Mode (SMD2 to SMD0 = 010b, IICM = 1)			
		IICM2 = 0 (ACK/NACK interrupt)		IICM2 = 1 (Transmit/receive interrupt)	
		CKPH = 0 (No clock delay)	CKPH = 1 (Clock delayed)	CKPH = 0 (No clock delay)	CKPH = 1 (Clock delayed)
Store received data	The first to eighth bits of received data are stored into bits 0 to 7 in the UiRB register	The first to eighth bits of received data are stored into bits 7 to 0 in the UiRB register		The first to seventh bits of received data are stored into bits 6 to 0 in the UiRB register and the eighth bit is stored into bit 8	Same as on the left column on the first data storing (2). The first to eighth bits of received data are stored into 7 to 0 bits in the UiRB register and the ninth bit is stored into bit 8 on the second data storing (3)
Read received data	The UiRB register status is read as it is			Bits 6 to 0 in the UiRB register are read as bits 7 to 1 and bit 8 is read as bit 0	Same as on the left column on the first read (2). The UiRB register status is read as it is on the second read (3)

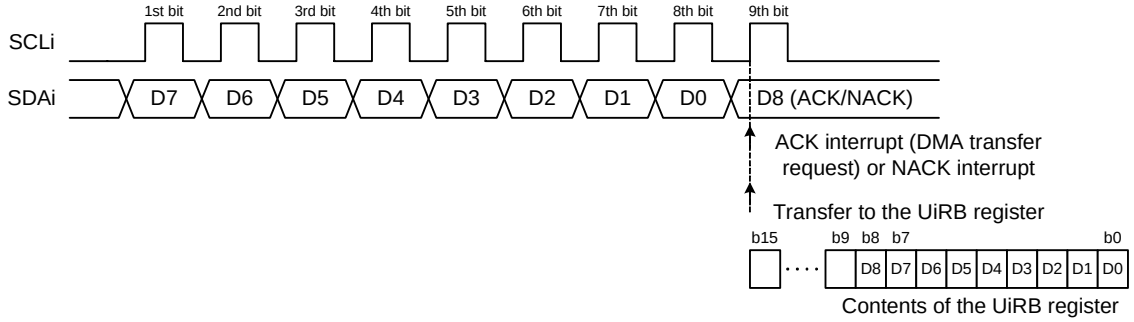
Notes:

1. Steps to change interrupt sources are as follows:
   (1) Disable the interrupt of the corresponding software interrupt number.
   (2) Change the source of interrupt.
   (3) Set the IR bit of the corresponding software interrupt number to 0 (no interrupt requested).
   (4) Set bits ILVL2 to ILVL0 of the corresponding software interrupt number.
2. The first data transfer to the UiRB register starts on the rising edge of the eighth bit of SCLi.
3. The second data transfer to the UiRB register starts on the rising edge of the ninth bit of SCLi.

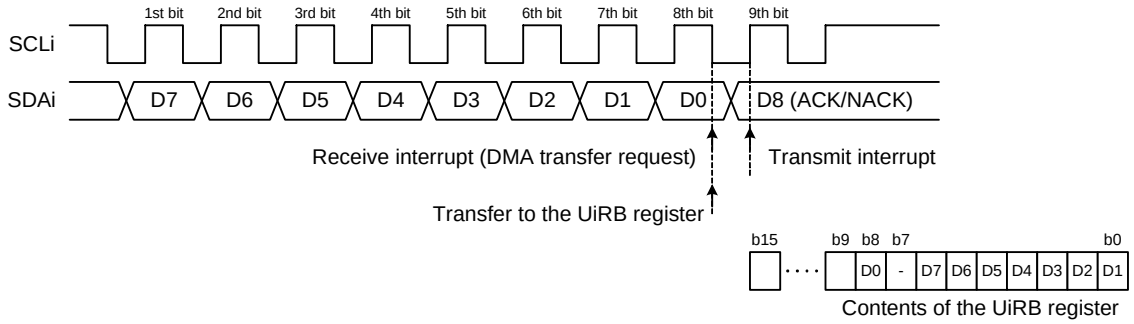
(1) When the IICM2 bit is set to 0 (Use ACK/NACK interrupt) and the CKPH bit is set to 0 (no clock delay)



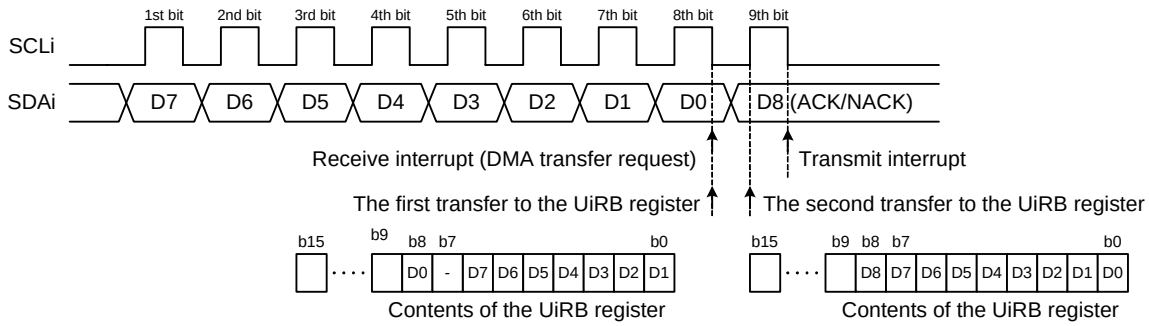
(2) When the IICM2 bit is set to 0 and the CKPH bit is set to 1 (clock delayed)



(3) When the IICM2 bit is set to 1 (Use transmit/receive interrupt) and the CKPH bit is set to 0



(4) When the IICM2 bit is set to 1 and the CKPH bit is set to 1



The figure above applies under the following condition:
-The CKDIR bit in the UiMR register is 0 (internal clock)

Figure 17.32 Timings for the Transfer and Interrupt to the UiRB Register (i = 0 to 6)

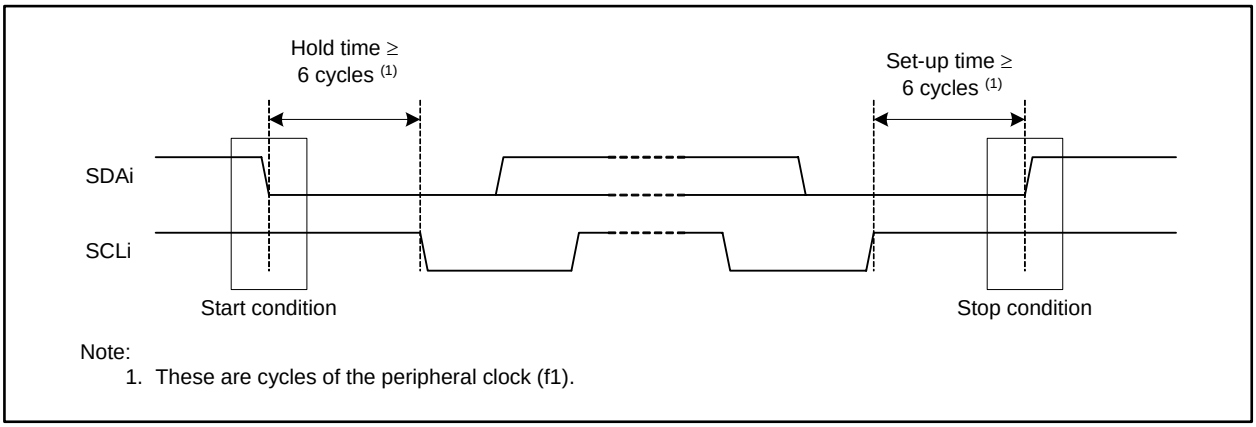
17.3.1 Start Condition and Stop Condition Detection

The start condition and stop condition are detected by their respective detectors.

The start condition detection interrupt request is generated by a high-to-low transition at the SDAi pin while the SCLi (i = 0 to 6) pin is held high. The stop condition detection interrupt request is generated by a low-to-high transition at the SDAi pin while the SCLi pin is held high.

The start condition detection interrupt shares interrupt control registers and vectors with the stop condition detection interrupt. The BBS bit in the UiSMR register determines which interrupt is requested.

To detect a start condition or stop condition, both set-up and hold times require six cycles or more of the peripheral clock (f1) as shown in Figure 17.33. To meet the condition for the Fast-mode specification, f1 is required to be 10 MHz or more.



Note:
1. These are cycles of the peripheral clock (f1).

Figure 17.33 Start Condition and Stop Condition Detection Timing (i = 0 to 6)

17.3.2 Start Condition and Stop Condition Generation

The start condition, restart condition, and stop condition are generated by bits STAREQ, RSTAREQ, and STPREQ in the UiSMR4 register (i = 0 to 6), respectively. To output the start condition, the STSPSEL bit in the UiSMR4 register should be set to 1 (start condition/stop condition generator selected) after setting the STAREQ bit to 1 (start). To output the restart condition and stop condition, the STSPSEL bit should be set to 1 after setting respective bits RSTAREQ and STPREQ to 1.

Table 17.12 and Figure 17.34 show the functions of the STSPSEL bit.

Table 17.12 STSPSEL Bit Functions

Function	STSPSEL = 0	STSPSEL = 1
Start condition and stop condition generation	Output is provided by the program with port (no auto generation by hardware)	Start condition or stop condition is output according to the STAREQ, RSTAREQ, or STPREQ bit, respectively
Start condition and stop condition interrupt request generating timing	When start condition or stop condition is detected	When start condition or stop condition generation is completed

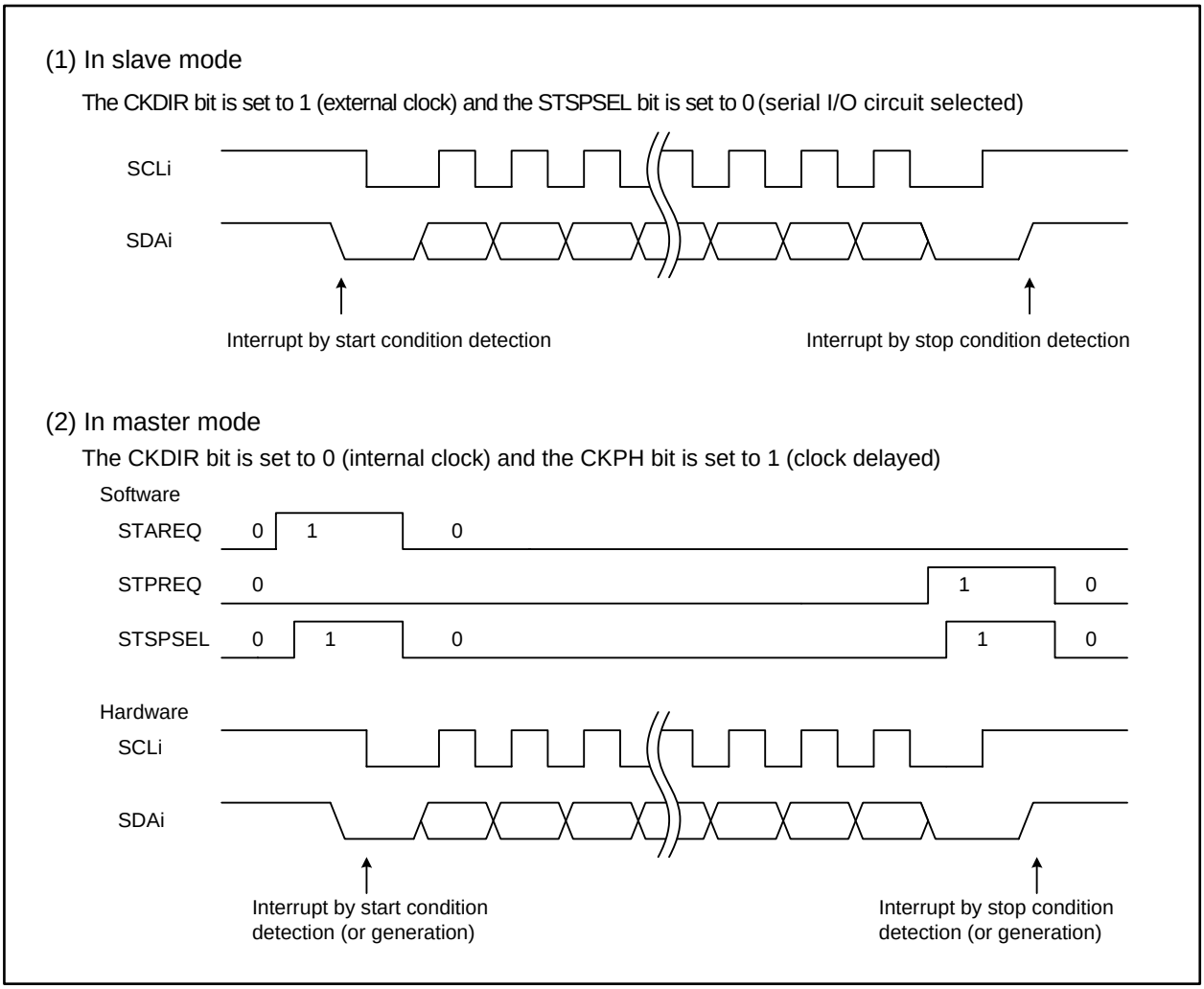

(1) In slave mode
The CKDIR bit is set to 1 (external clock) and the STSPSEL bit is set to 0 (serial I/O circuit selected)
SCLi
SDAi
Interrupt by start condition detection
Interrupt by stop condition detection
(2) In master mode
The CKDIR bit is set to 0 (internal clock) and the CKPH bit is set to 1 (clock delayed)
Software
STAREQ
STPREQ
STSPSEL
Hardware
SCLi
SDAi
Interrupt by start condition detection (or generation)
Interrupt by stop condition detection (or generation)

Figure 17.34 STSPSEL Bit Functions (i = 0 to 6)

17.3.3 Arbitration

The MCU determines whether the transmit data matches data input to the SDAi pin on the rising edge of the SCLi. If it does not match the input data, the arbitration takes place at the SDAi pin by switching off the data output stage.

The ABC bit in the UiSMR register (i = 0 to 6) determines the update timing for the ABT bit in the UiRB register.

When the ABC bit is set to 0 (update per bit), the ABT bit is set to 1 (arbitration is lost) as soon as a data discrepancy is detected. If not detected, the ABT bit is set to 0 (arbitration is won). When the ABC bit is set to 1 (update per byte), the ABT bit is set to 1 on the falling edge of the eighth bit of the SCLi if any discrepancy is detected. In this ABC bit setting, the ABT bit should be set to 0 to start the next 1-byte transfer.

When the ALS bit in the UiSMR2 register is set to 1 (SDA output stop enabled), an arbitration lost occurs. As soon as the ABT bit is set to 1, the SDAi pin becomes high-impedance.

17.3.4 SCL Control and Clock Synchronization

Data transmission/reception in I^2C mode uses the transmit/receive clock as shown in Figure 17.32. The clock speed increase makes it difficult to secure the required time for ACK generation and data transmit procedure. The I^2C mode supports a function of wait-state insertion to secure this required time and a function of clock synchronization with a wait-state inserted by other devices.

The SWC bit in the UiSMR2 register (i = 0 to 6) is used to insert a wait-state for ACK generation. When the SWC bit is set to 1 (hold the SCLi pin low after the eighth bit is received), the SCLi pin is held low on the falling edge of the eighth bit of the SCLi. When the SWC bit is set to 0 (no wait-state/wait-state cleared), the SCLi line is released.

When the SWC2 bit in the UiSMR2 register is set to 1 (hold the SCLi pin low), the SCLi pin is forced low even during transmission or reception in progress. When the SWC2 bit is set to 0 (output the transmit/receive clock at the SCLi pin), the SCLi line is released to output the transmit/receive clock.

The SWC9 bit in the UiSMR4 register is used to insert a wait-state for checking received acknowledge bits. While the CKPH bit in the UiSMR3 register is set to 1 (clock delayed), when the SWC9 bit is set to 1 (hold the SCLi pin low after the ninth bit is received), the SCLi pin is held low on the falling edge of the ninth bit of the SCLi. When the SWC9 bit is set to 0 (no wait-state/wait-state cleared), the SCLi line is released.

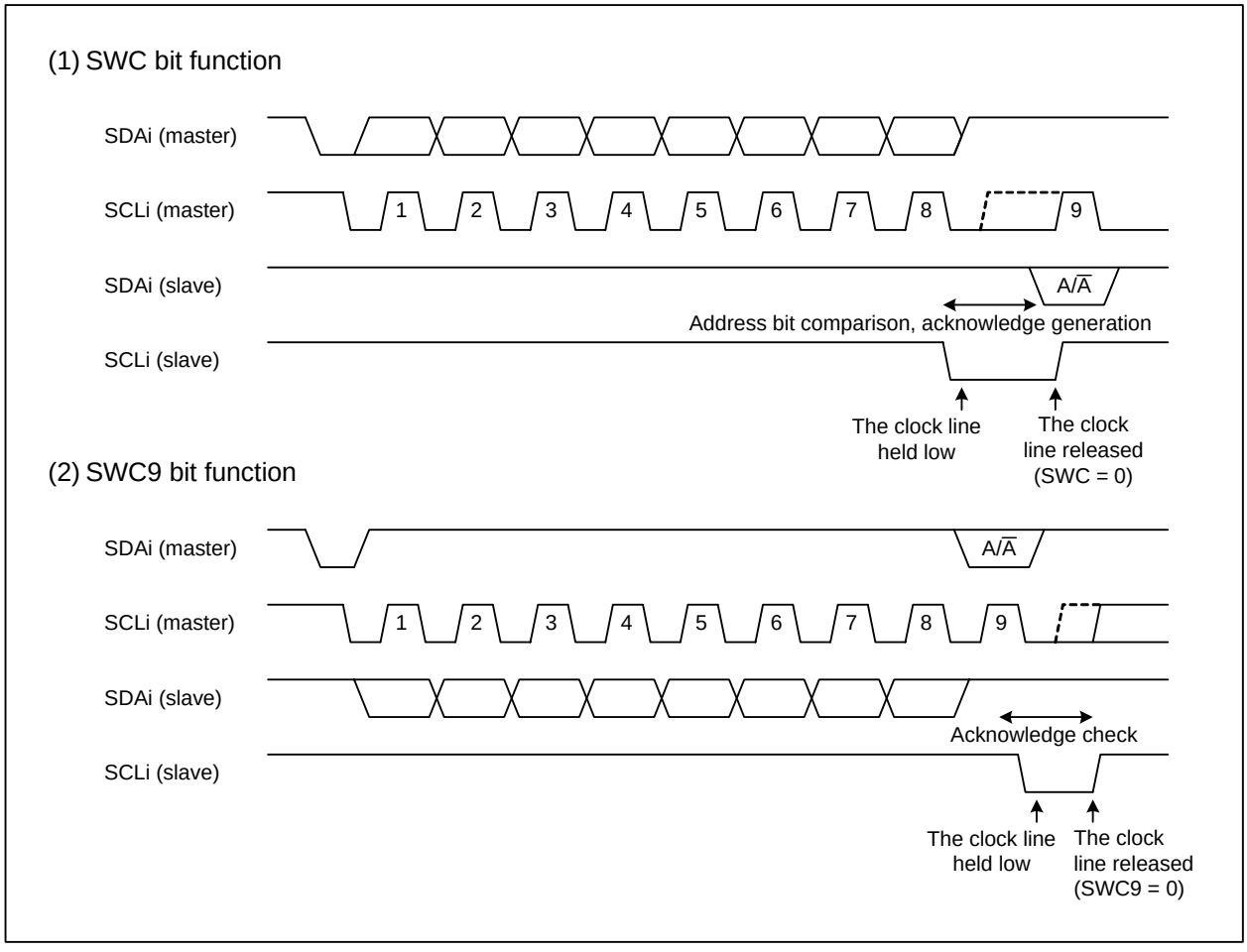


Figure 17.35 Wait-state Insertion by Bits SWC or SWC9 (i = 0 to 6)

The CSC bit in the UiSMR2 register is to synchronize an internally generated clock with the clock applied to the SCLi pin. For example, if a wait-state is inserted from other devices, the two clocks are not synchronized. While the CSC bit is set to 1 (clock synchronization enabled) and the internal clock is held high, when a high at the SCLi pin changes to low, the internal clock becomes low in order to reload the value of the UiBRG register and to resume counting. While the SCLi pin is held low, when the internal clock changes from low to high, the count is stopped until the SCLi pin becomes high. That is, the UARTi transmit/receive clock is the logical AND of the internal clock and the SCLi. The synchronized period starts from one clock prior to the first synchronized clock and ends when the ninth clock is completed. The CSC bit can be set to 1 only when the CKDIR bit in the UiMR register is set to 0 (internal clock selected).

The SCLHI bit in the UiSMR4 register is used to leave the SCLi pin open when other master generates a stop condition while the master is in transmit/receive operation. If the SCLHI bit is set to 1 (output stopped), the SCLi pin is open (the pin is high-impedance) when a stop condition is detected and the clock output is stopped.

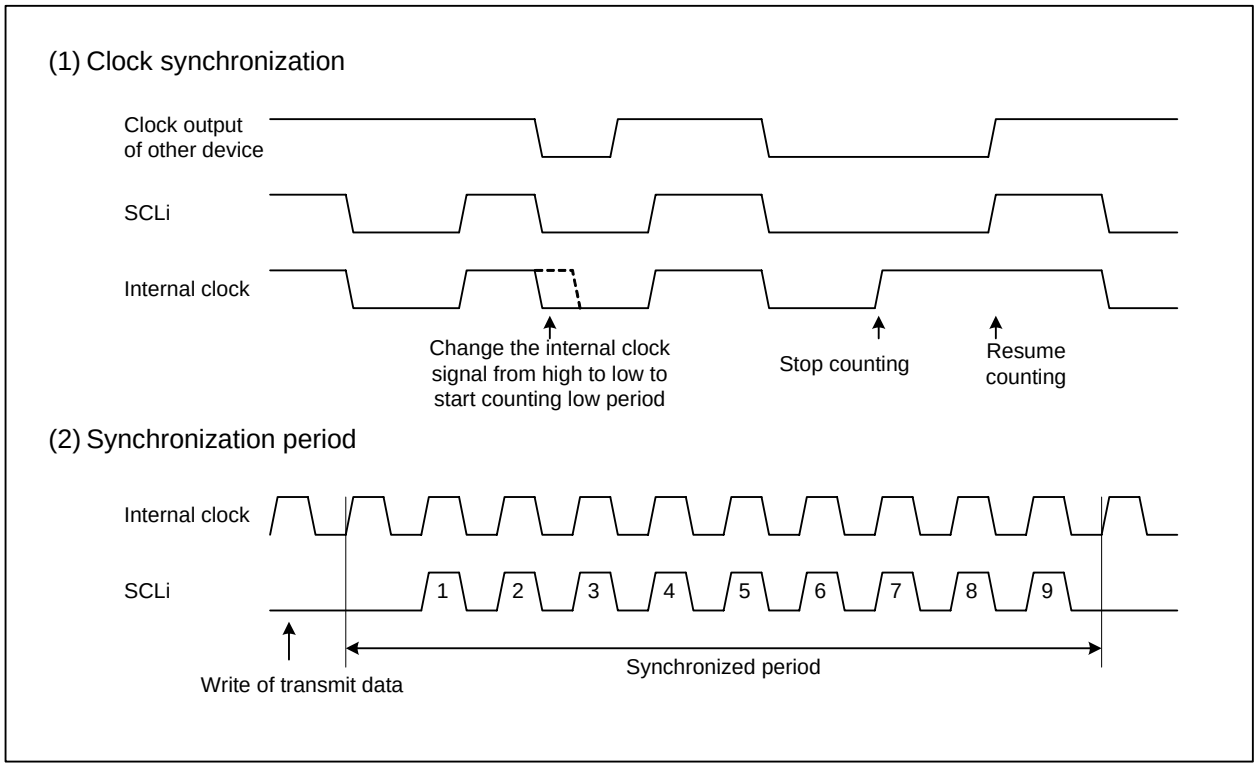


Figure 17.36 Clock Synchronization (i = 0 to 6)

17.3.5 SDA Output

Values set in bits 8 to 0 (D8 to D0) in the UiTB register (i = 0 to 6) are output starting from D7 to D0, and lastly D8, which is a bit for the acknowledge signal. When transmitting, D8 should be set to 1 to free the bus. When receiving, D8 should be set to ACK or NACK.

Bits DL2 to DL0 in the UiSMR3 register set a delay time of the SDAi on the falling edge of the SCLi. Based on the UiBRG count source, the delay time can be selected from zero cycles (no delay) and two to eight cycles.

When the SDHI bit in the UiSMR2 register is set to 1 (SDA output disabled), the SDAi pin forcibly becomes high-impedance. Output at the SDAi pin is low if the I/O port is selected for the SDAi and the pin is specified as the output port after selecting I^2C mode. In this case, if the SDHI bit is set to 1, the SDAi pin becomes high-impedance.

When the SDHI bit is rewritten while the SCLi pin is held high, a start condition or stop condition is generated. When it is rewritten immediately before the rising edge of the SCLi, an arbitration lost may be accidently detected. Therefore, the SDHI bit should be rewritten so the SDAi pin level changes while the SCLi pin is low.

17.3.6 SDA Input

When the IICM2 bit in the UiSMR2 register (i = 0 to 6) is set to 0, the first eight bits of received data (D7 to D0) are stored into bits 7 to 0 in the UiRB register and the ninth bit (ACK/NACK) is stored into bit 8.

When the IICM2 bit is set to 1, the first seven bits of received data (D7 to D1) are stored into bits 6 to 0 in the UiRB register and eighth bit (D0) is stored into bit 8.

If the IICM2 bit is set to 1 and the CKPH bit in the UiSMR3 register is set to 1 (clock delayed), the same data that is set when the IICM2 bit is 0 can be read. To read this data, read the UiRB register after data in the ninth bit is latched on the rising edge of the SCLi.

17.3.7 Acknowledge

When data is to be received in master mode, the ACK is output after eight bits are received by setting the UiTB register to 00FFh as dummy data. When the STSPSEL bit in the UiSMR4 register (i = 0 to 6) is set to 0 (serial I/O circuit selected) and the ACKC bit is set to 1 (ACK data output), the value of the ACKD bit is output at the SDAi pin.

If the IICM2 bit is set to 0, the NACK interrupt request is generated when the SDAi pin is held high on the rising edge of the ninth bit of the SCLi. The ACK interrupt request is generated when the SDAi pin is held low.

If the DMA request source is “UARTi receive interrupt request or ACK interrupt request”, the DMA transfer is activated when an ACK is detected.

17.3.8 Initialization of Transmit/Receive Operation

When the CKDIR bit in the UiMR register (i = 0 to 6) is set to 1 (external clock) and the STC bit in the UiSMR2 register is set to 1 (the circuit is initialized), and a start condition is detected, the following three operations are performed:

- The transmit register is reset and the content of the UiTB register is transferred to the transmit register. The new data transmission starts on the falling edge of the first bit of the next SCLi as transmit clock. The content of the transmit register before the reset is output at the SDAi pin in the period from the falling edge of the SCLi until the first data output.
- The receive register is reset and the new data reception starts on the falling edge of the first bit of the next SCLi.
- The SWC bit in the UiSMR2 register is set to 1 (hold the SCLi pin low after the eighth bit is received).

If UARTi transmission/reception is started with this function, the TI bit in the UiC1 register does not change.

17.4 Special Mode 2

Special mode 2 enables serial communication between one or multiple masters and multiple slaves. The $\overline{SSi}$ input pin (i = 0 to 6) controls the serial bus communication. Table 17.13 lists specifications of special mode 2.

Table 17.13 Special Mode 2 Specifications

Item	Specification
Data format	8-bit character length
Transmit/receive clock	• The CKDIR bit in the UiMR register (i = 0 to 6) is set to 0 (internal clock): $\frac{fx}{2(m+1)}$ fx = f1, f8, f2n m: UiBRG register setting value, 00h to FFh • The CKDIR bit is set to 1 (external clock): input into the CLKi pin
Transmit/receive control	SS function
Transmit start conditions	The conditions for starting data transmission are as follows (1): • The TE bit in the UiC1 register is set to 1 (transmission enabled) • The TI bit in the UiC1 register is set to 0 (data held in the UiTB register)
Receive start conditions	The conditions for starting data reception are as follows (1): • The RE bit in the UiC1 register is set to 1 (reception enabled) • The TE bit in the UiC1 register is set to 1 (transmission enabled) • The TI bit in the UiC1 register is set to 0 (data held in the UiTB register)
Interrupt request generating timing	In transmit interrupt, one of the following conditions is selected to set the UiIRS bit in registers U0C1 to U6C1: • The UiIRS bit is set to 0 (transmit buffer in the UiTB register is empty): when data is transferred from the UiTB register to the UARTi transmit register (when the transmission has started) • The UiIRS bit is set to 1 (transmission is completed): when data transmission from the UARTi transmit register is completed In receive interrupt, • When data is transferred from the UARTi receive register to the UiRB register (when the reception is completed)
Error detection	Overrun error (2) This error occurs when the seventh bit of the next data has been received before reading the UiRB register
Selectable functions	• CLK polarity Selectable either rising or falling edge of the transmit/receive clock for transfer data input and output • Bit order selection Selectable either LSB first or MSB first • Continuous receive mode Data reception is enabled by a read access to the UiRB register • Serial data logical inversion This function logically inverses transmit/receive data • Clock phase selection Selectable from one of four combinations of transmit/receive clock polarity and phases • $\overline{SSi}$ input pin function Output pin can be high-impedance when the $\overline{SSi}$ pin is high

Notes:
1. In case external clock is selected, the following preconditions should be met:
   • The CLKi pin is held high when the CKPOL bit in the UiC0 register is set to 0 (transmit data output on the falling edge of the transmit/receive clock and receive data input on the rising edge)
   • The CLKi pin is held low when the CKPOL bit is set to 1 (transmit data output on the rising edge of the transmit/receive clock and receive data input on the falling edge)
2. If an overrun error occurs, the UiRB register is undefined. The IR bit in the SiRIC register is not changed to 1 (interrupts requested).

Table 17.14 lists register settings in special mode 2.

Table 17.14 Register Settings in Special Mode 2 (i = 0 to 6)

Register	Bits	Function
UiMR	7 to 4	Set the bits to 0000b
	CKDIR	Set the bit to 0 in master mode and set it to 1 in slave mode
	SMD2 to SMD0	Set the bits to 001b
UiC0	UFORM	Select either LSB first or MSB first
	CKPOL	Clock phase can be set by the combination of bits CKPOL and CKPH in the UiSMR3 register
	5	Set the bit to 0
	CRD	Set the bit to 1
	TXEPT	Transmit register empty flag
	2	Set the bit to 0
	CLK1 and CLK0	Select a count source for the UiBRG register
UiC1	7 and 6	Set the bits to 00b
	UiRRM	Set the bit to 1 to use continuous receive mode
	UiIRS	Select a source for UARTi transmit interrupt
	RI	Receive complete flag
	RE	Set the bit to 1 to enable data reception
	TI	Transmit buffer empty flag
	TE	Set the bit to 1 to enable data transmission/reception
UiSMR	7 to 0	Set the bits to 00h
UiSMR2	7 to 0	Set the bits to 00h
UiSMR3	7 to 5	Set the bits to 000b
	ERR	Mode fault flag
	3	Set the bit to 0
	DINC	Set to 0 in master mode and set to 1 in slave mode
	CKPH	Clock phase can be set by the combination of bits CKPH and CKPOL in the UiC0 register
	SSE	Set the bit to 1
UiSMR4	7 to 0	Set the bits to 00h
UiBRG	7 to 0	Set the bit rate
IFS0	IFS06	Select input pins for CLK3, RXD3, SRXD3, and $\overline{SS3}$
	IFS03 and IFS02	Select input pins for CLK6, RXD6, SRXD6, and $\overline{SS6}$
UiTB	7 to 0	Set the data to be transmitted
UiRB	OER	Overrun error flag
	7 to 0	Received data is read

17.4.1 $\overline{SSi}$ Input Pin Function (i = 0 to 6)

Special mode 2 is selected by setting the SSE bit in the UiSMR3 register to 1 (SS enabled). The $\overline{CTSi}$/$\overline{RTSi}$/$\overline{SSi}$ pin functions as $\overline{SSi}$ input.

The DINC bit in the UiSMR3 register determines which MCU performs as master or slave.

When multiple MCUs perform as master (multi-master system), the $\overline{SSi}$ pin setting determines which master MCU is active and when.

17.4.1.1 SS Function in Slave Mode

When the DINC bit is set to 1 (slave mode selected) while an input at the $\overline{SSi}$ pin is high, the STXDi pin becomes high-impedance and the clock input at the CLKi pin is ignored. When an input at the $\overline{SSi}$ pin is low, the clock input is valid and serial data is output from the STXDi pin to enable serial communication.

17.4.1.2 SS Function in Master Mode

When the DINC bit is set to 0 (master mode selected) while an input at the $\overline{SSi}$ pin is high, which means there is the only one master MCU or no other master MCU is active, the MCU as master starts communication. The master provides the transmit/receive clock output at the CLKi pin. When input at the $\overline{SSi}$ pin is low, which means that there are more masters, pins TXDi and CLKi become high-impedance. This error is called a mode fault. It can be verified using the ERR bit in the UiSMR3 register. The ongoing data transmission/reception is not stopped even if a mode fault occurs. To stop transmission/reception, bits SMD2 to SMD0 in the UiMR register should be set to 000b (serial interface disabled).

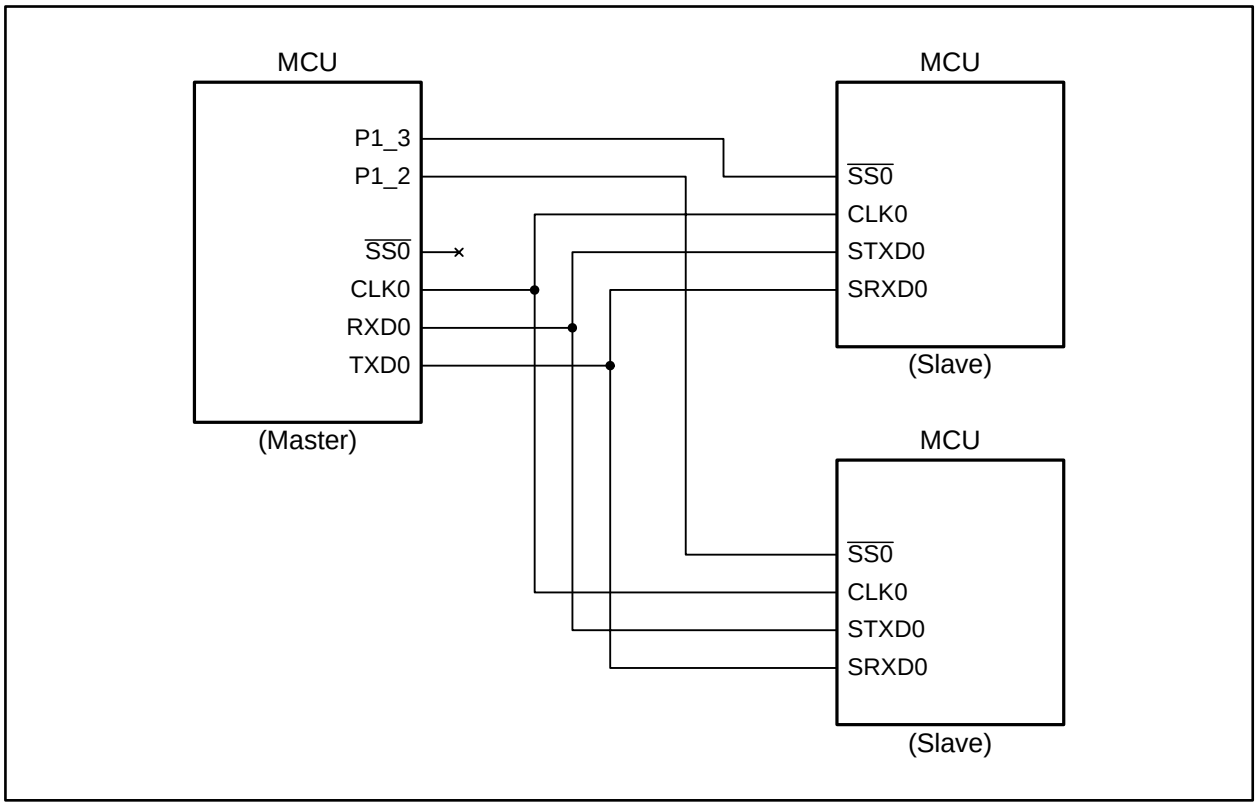


Figure 17.37 Serial Bus Communication Control with $\overline{SSi}$ Pin

17.4.2 Clock Phase Setting

The CKPH bit in the UiSMR3 register (i = 0 to 6) and the CKPOL bit in the UiC0 register select one of four combinations of transmit/receive clock polarity and serial clock phase.
The transmit/receive clock phase and polarity should be identical for the master device and the communicating slave device.

17.4.2.1 Transmit/Receive Timing in Master Mode

When the DINC bit is set to 0 (master mode selected), the CKDIR bit in the UiMR register should be set to 0 (internal clock) to generate the clock. Figure 17.38 shows transmit/receive timing of each clock phase.

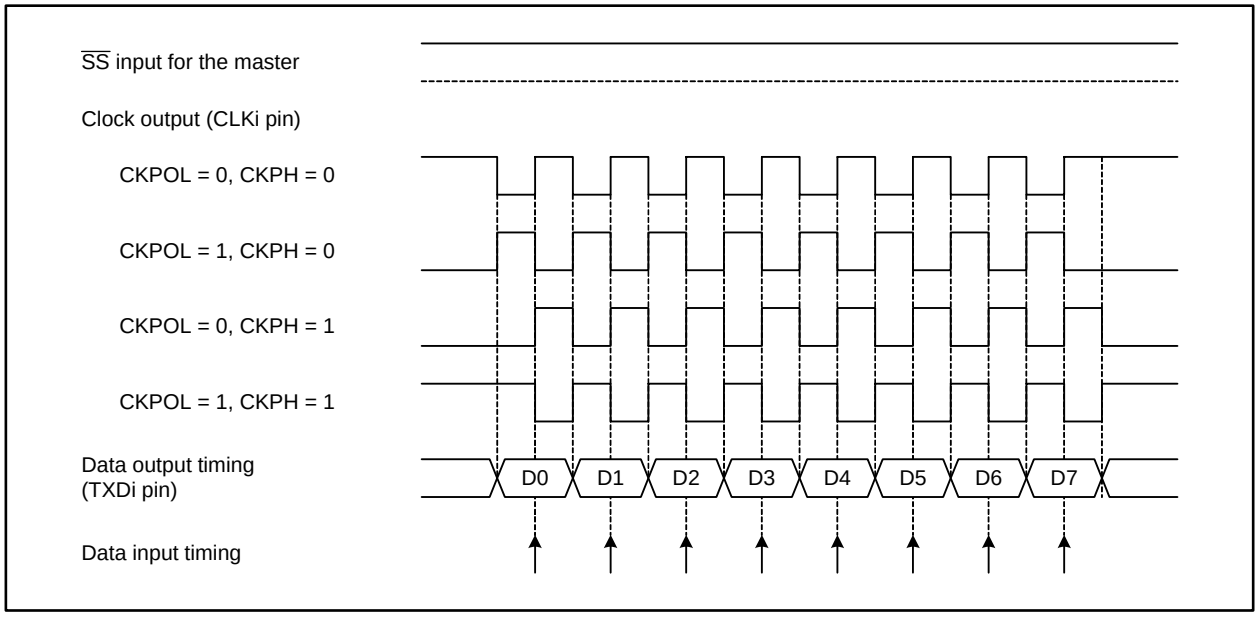


Figure 17.38 Transmit/Receive Timing in Master Mode

17.4.2.2 Transmit/Receive Timing in Slave Mode

When the DINC bit is set to 1 (slave mode selected), the CKDIR bit in the UiMR register should be set to 1 (external clock).

When the CKPH bit is set to 0 (no clock delay) while input at the $\overline{SSi}$ pin is high, the STXDi pin becomes high-impedance. When input at the $\overline{SSi}$ pin is low, the conditions for data transmission are all met, but output is undefined. Then the data transmission/reception starts synchronizing with the clock. Figure 17.39 shows the transmit/receive timing.

When the CKPH bit is set to 1 (clock delayed) while an input at the $\overline{SSi}$ pin is high, the STXDi pin becomes high-impedance. When an input at the $\overline{SSi}$ pin is low, the first data is output. Then the data transmission starts synchronizing with the clock. Figure 17.40 shows the transmit/receive timing.

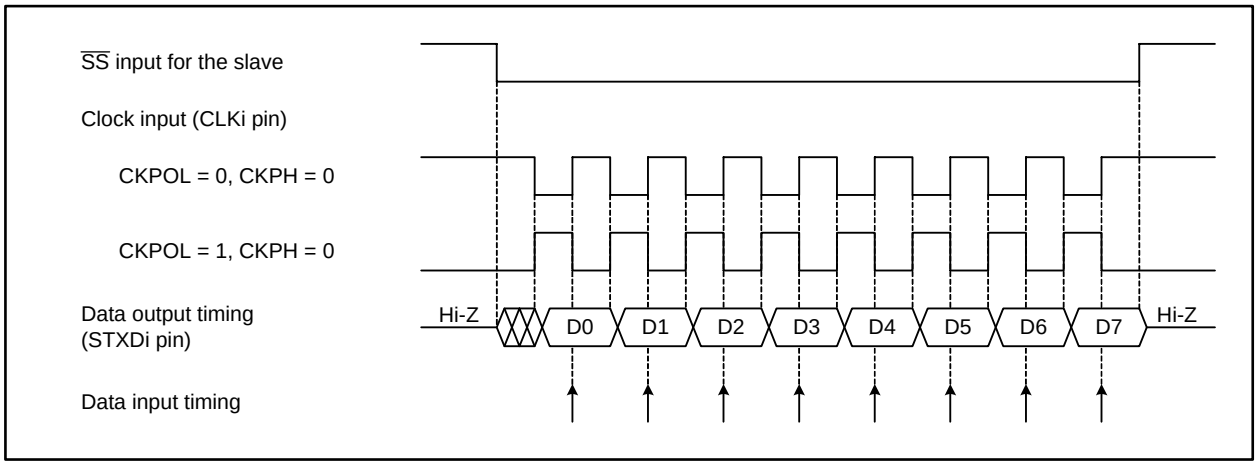


Figure 17.39 Transmit/Receive Timing in Slave Mode (CKPH = 0)

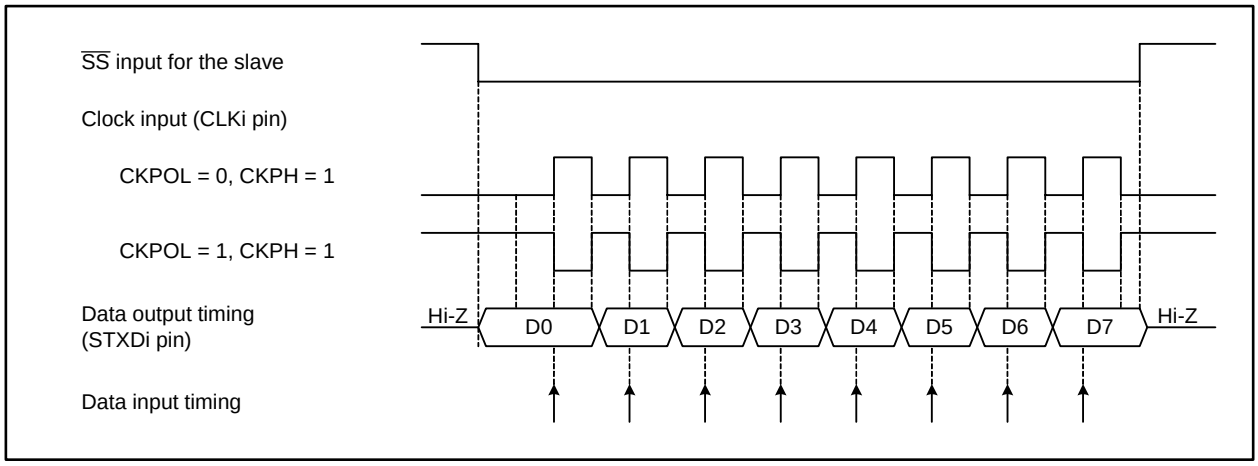


Figure 17.40 Transmit/Receive Timing in Slave Mode (CKPH = 1)

17.5 Notes on Serial Interface

17.5.1 Changing the UiBRG Register (i = 0 to 8)

- Set the UiBRG register after setting bits CLK1 and CLK0 in the UiC0 register. When these bits are changed, the UiBRG register must be set again.
- If a clock is input immediately after the UiBRG register is set to 00h, the counter reaches FFh. In this case, it requires an extra 256 clocks to reload 00h into the register. Once the 00h is reloaded, the counter performs the operation without dividing the count source according to the setting.

17.5.2 Synchronous Serial Interface Mode

17.5.2.1 Selecting an External Clock

- If an external clock is selected, the following conditions must be met while the external clock is held high when the CKPOL bit in the UiC0 register (i = 0 to 8) is set to 0 (transmit data output on the falling edge of the transmit/receive clock and receive data input on the rising edge), or while the external clock is held low when the CKPOL bit is set to 1 (transmit data output on the rising edge of the transmit/receive clock and receive data input on the falling edge):
  - The TE bit in the UiC1 register is set to 1 (transmission enabled).
  - The RE bit in the UiC1 register is set to 1 (reception enabled). This bit setting is not required in transmit operation only.
  - The TI bit in the UiC1 register is set to 0 (data held in the UiTB register).

17.5.2.2 Receive Operation

- In synchronous serial interface mode, the transmit/receive clock is controlled by the transmit control circuit. Set the UARTi-associated registers (i = 0 to 8) for a transmit operation, even if the MCU is used only for receive operation. Dummy data is output from the TXDi pin while receiving if the TXDi pin is set to output mode.

- If data is received continuously, an overrun error occurs when the RI bit in the UiC1 register is 1 (data held in the UiRB register) and the seventh bit of the next data is received in the UARTi receive shift register. Then, the OER bit in the UiRB register becomes 1 (overrun error occurred). In this case, the UiRB register becomes undefined. If an overrun error occurs, the IR bit in the SiRIC register is not changed to 1.

17.5.3 Special Mode 1 (I^2C Mode)

- To generate a start condition, stop condition, or restart condition, set the STSPSEL bit in the UiSMR4 register (i = 0 to 6) to 0. Then, wait a half or more clock cycles of the transmit/receive clock to change the respective condition generate bit (the STAREQ, RSTAREQ, or STPREQ bit) from 0 to 1.

17.5.4 Reset Procedure on Communication Error

- Operations which result in communication errors such as rewriting function select registers during transmission/reception should not be performed. Follow the procedure below to reset the internal circuit once the communication error occurs in the following cases: when the operation above is performed by a receiver or transmitter or when a bit slip is caused by noise.

A. Synchronous Serial Interface Mode

(1) Set the TE bit in the UiC1 register (i = 0 to 8) to 0 (transmission disabled) and the RE bit to 0 (reception disabled).
(2) Set bits SMD2 to SMD0 in the UiMR register to 000b (serial interface disabled).
(3) Set bits SMD2 to SMD0 in the UiMR register to 001b (synchronous serial interface mode).
(4) Set the TE bit in the UiC1 register to 1 (transmission enabled) and the RE bit to 1 (reception enabled) if necessary.

B. UART Mode

(1) Set the TE bit in the UiC1 register to 0 (transmission disabled) and the RE bit to 0 (reception disabled).
(2) Set bits SMD2 to SMD0 in the UiMR register to 000b (serial interface disabled).
(3) Set bits SMD2 to SMD0 in the UiMR register to 100b (UART mode, 7-bit character length), 101b (UART mode, 8-bit character length), or 110b (UART mode, 9-bit character length).
(4) Set the TE bit in the UiC1 register to 1 (transmission enabled) and the RE bit to 1 (reception enabled) if necessary.

18. A/D Converter

The A/D converter consists of one 10-bit successive approximation A/D converter with a capacitive coupling amplifier.

The result of an A/D conversion is stored in the A/D registers corresponding to selected pins. It is stored in the AD00 register only when DMAC operating mode is enabled.

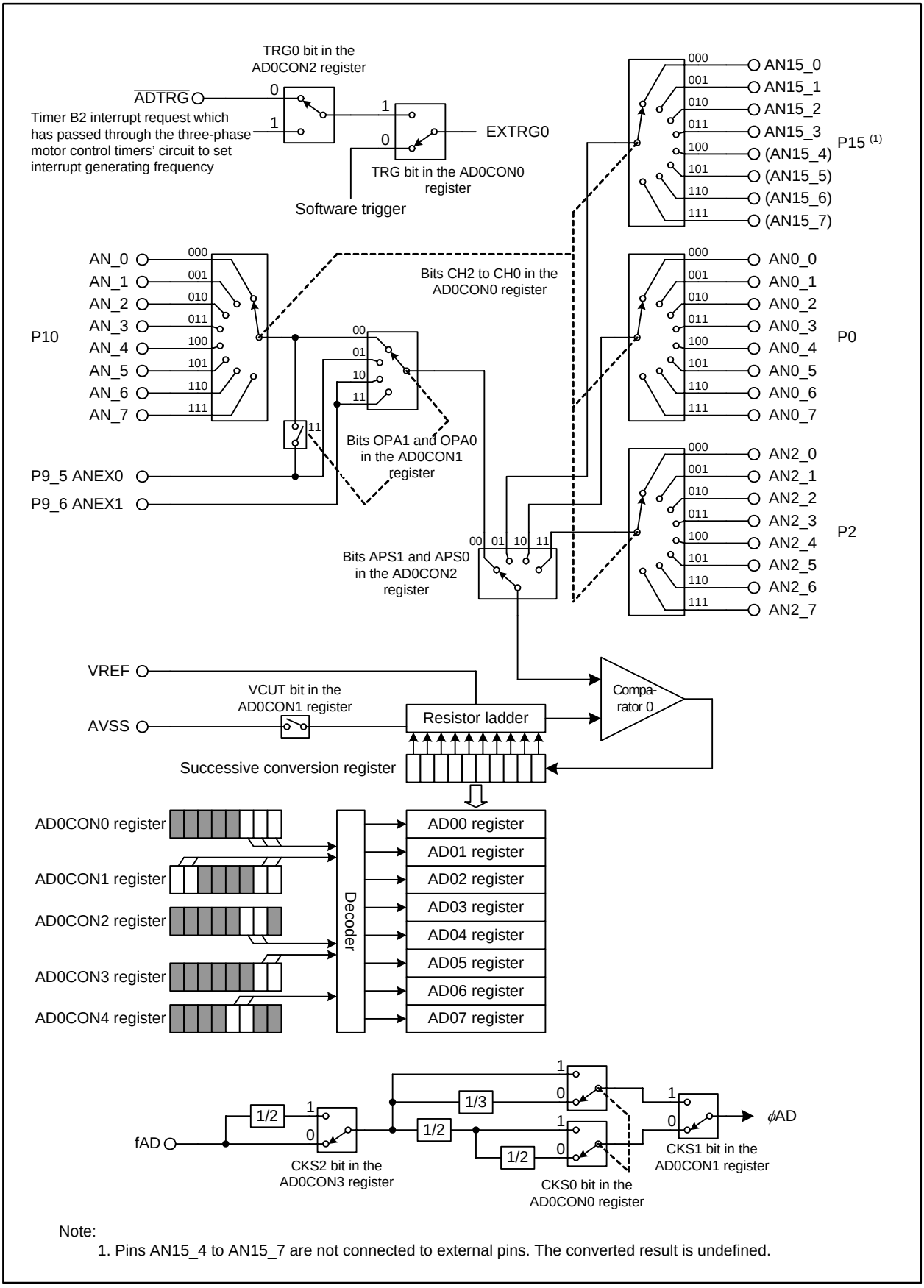
When the A/D converter is not in use, power consumption can be reduced by setting the VCUT bit in the AD0CON1 register to 0 (VREF disconnected). This bit setting enables the power supply from VREF pin to the resistor ladder to stop.

Table 18.1 lists specifications of the A/D converter. Figure 18.1 shows a block diagram of the A/D converter. Figure 18.2 to Figure 18.7 show registers associated with the A/D converter.

Table 18.1 A/D Converter Specifications

Item	Specification
A/D conversion method	Capacitance-based successive approximation
Analog input voltage (1)	0 V to AVCC (VCC)
Operating clock, ϕAD (2)	fAD, fAD/2, fAD/3, fAD/4, fAD/6, or fAD/8
Resolution	8 bits or 10 bits
Operating modes	One-shot mode, repeat mode, single sweep mode, repeat sweet mode 0, repeat sweep mode 1, multi-port single sweep mode, multi-port repeat sweep mode 0
Analog input pins (3)	30 8 pins each for AN, AN0, and AN2, 4 pins for AN15 2 function-extended input pins (ANEX0 and ANEX1)
A/D conversion start conditions	• Software trigger The ADST bit in the AD0CON0 register is set to 1 (A/D conversion started) by a program • External trigger (Retrigger is enabled) An input signal at the $\overline{\text{ADTRG}}$ pin switches from high to low after the ADST bit is set to 1 by a program • Hardware trigger (Retrigger is enabled) • Generation of a timer B2 interrupt request which has passed through the circuit to set interrupt generating frequency in the three-phase motor control timers after the ADST bit is set to 1 by a program
Conversion rates per pin	• Without sample and hold function 49 ϕAD cycles @ 8-bit resolution 59 ϕAD cycles @ 10-bit resolution including 2 ϕAD cycles for sampling time • With sample and hold function 28 ϕAD cycles @ 8-bit resolution 33 ϕAD cycles @ 10-bit resolution including 3 ϕAD cycles for sampling time

Notes:
1. Analog input voltage is not affected by with/without the sample and hold function.
2. The ϕAD frequency should be as follows:
   • 10 MHz or below
   • Without the sample and hold function, 250 kHz or above
   • With the sample and hold function, 1 MHz or above
3. When AVCC = VREF = VCC, A/D input voltage for pins AN_0 to AN_7, AN0_0 to AN0_7, AN2_0 to AN2_7, AN15_0 to AN15_3, ANEX0, and ANEX1 should be VCC or lower.



Note:
1. Pins AN15_4 to AN15_7 are not connected to external pins. The converted result is undefined.

Figure 18.1 A/D Converter Block Diagram

A/D0 Control Register 0 (1)

b7 b6 b5 b4 b3 b2 b1 b0

Symbol	Address	Reset Value
AD0CON0	0396h	0000 0000b

Bit Symbol	Bit Name	Function	RW
CH0	Analog Input Pin Select Bit (2, 3, 4)	b2 b1 b0 0 0 0 : ANi_0 0 0 1 : ANi_1 0 1 0 : ANi_2 0 1 1 : ANi_3 1 0 0 : ANi_4 (5) 1 0 1 : ANi_5 (5) 1 1 0 : ANi_6 (5) 1 1 1 : ANi_7 (5) (i = none, 0, 2, 15)	RW
CH1			RW
CH2			RW
MD0	A/D Operating Mode Select Bit 0 (2, 6, 7)	b4 b3 0 0 : One-shot mode 0 1 : Repeat mode 1 0 : Single sweep mode 1 1 : Repeat sweep mode 0 or 1	RW
MD1			RW
TRG	Trigger Select Bit	0: Software trigger 1: External trigger or hardware trigger (8)	RW
ADST	A/D Conversion Start Bit	0: A/D conversion stopped 1: A/D conversion started (8)	RW
CKS0	Frequency Select Bit	(Note 9)	RW

Notes:

1. When this register is rewritten during an A/D conversion, the converted result is undefined.
2. Analog input pins should be set again after the A/D operating mode is changed.
3. This bit setting is enabled in one-shot mode or repeat mode.
4. Either AN, AN0, AN2, or AN15 port should be selected by using bits APS1 and APS0 in the AD0CON2 register.
5. Do not select pins AN15_4 to AN15_7.
6. When the MSS bit in the AD0CON3 register is set to 1 (multi-port sweep mode enabled), bits MD1 and MD0 should be set to 10b for multi-port single sweep mode and 11b for multi-port repeat sweep mode 0.
7. When the MSS bit in the AD0CON3 register is set to 1, bits MD1 and MD0 should be set to 10b or 11b.
8. To use the external trigger or the hardware trigger, a source of trigger should be selected using the TRG0 bit in the AD0CON2 register, the TRG bit should be set to 1, then the ADST bit should be set to 1.
9. The ϕAD frequency should be 10 MHz or below. It is selected from the combination of bits CKS0, CKS1, and CKS2 shown as below:

The CKS2 bit in the AD0CON3 register	The CKS0 bit in the AD0CON0 register	The CKS1 bit in the AD0CON1 register	ϕAD
0	0	0	fAD divided by 4
		1	fAD divided by 3
	1	0	fAD divided by 2
		1	fAD
1	0	0	fAD divided by 8
		1	fAD divided by 6

Figure 18.2 AD0CON0 Register

A/D0 Control Register 1 (1)

b7 b6 b5 b4 b3 b2 b1 b0

Symbol	Address	Reset Value
AD0CON1	0397h	0000 0000b

Bit Symbol	Bit Name	Function	RW
SCAN0	A/D Sweep Pin Select Bit (2, 3, 4)	In single sweep mode or repeat sweep mode 0 b1 b0 0 0 : ANi_0, ANi_1 0 1 : ANi_0 to ANi_3 1 0 : ANi_0 to ANi_5 1 1 : ANi_0 to ANi_7 In repeat sweep mode 1 (5) b1 b0 0 0 : ANi_0 0 1 : ANi_0, ANi_1 1 0 : ANi_0 to ANi_2 1 1 : ANi_0 to ANi_3 In multi-port single sweep mode or multi-port repeat sweep mode 0 b1 b0 1 1 : ANi_0 to ANi_7 (6) (i = none, 0, 2, 15)	RW
SCAN1			RW
MD2	A/D Operating Mode Select Bit 1	0: Mode other than repeat sweep mode 1 1: Repeat sweep mode 1(7)	RW
BITS	8/10-bit Mode Select Bit	0: 8-bit mode 1: 10-bit mode	RW
CKS1	Frequency Select Bit	(Note 8)	RW
VCUT	VREF Connection Bit (9)	0: VREF disconnected (10) 1: VREF connected (11)	RW
OPA0	External Op-Amp Connect Mode Bit (12, 13)	b7 b6 0 0 : No use of ANEX0 or ANEX1 pin (Convert input at pins ANi_0 to ANi_7) 0 1 : Convert input at the ANEX0 pin 1 0 : Convert input at the ANEX1 pin 1 1 : External op-amp connected	RW
OPA1			RW

Notes:
1. When this register is rewritten during an A/D conversion, the converted result is undefined.
2. This bit setting is enabled in single sweep mode, repeat sweep mode 0, repeat sweep mode 1, multi-port single sweep mode, or multi-port repeat sweep mode 0.
3. Either AN, AN0, AN2, or AN15 port should be selected by using bits APS1 and APS0 in the AD0CON2 register.
4. The converted result of signals AN15_4 to AN15_7 is undefined. The bit setting should not include these signals.
5. These pins are commonly used in the A/D conversion when the MD2 bit is set to 1.
6. Set bits SCAN0 and SCAN1 to 11b in multi-port single sweep mode or multi-port repeat sweep mode 0.
7. When the MSS bit in the AD0CON3 register is set to 1 (multi-port sweep mode enabled), the MD2 bit should be set to 0.
8. Refer to the note on the CKS0 bit in the AD0CON0 register.
9. This bit controls the reference voltage to the A/D converter. It does not affect the VREF performance of the D/A converter.
10. The VCUT bit should not be set to 0 during the A/D conversion.
11. When the VCUT bit is switched from 0 to 1, the A/D conversion should be started after 1 µs or more.
12. Bits OPA 0 and OPA1 can be set to 01b or 10b only in one-shot mode or repeat mode. They should be set to 00b or 11b in other modes.
13. When the MSS bit in the AD0CON3 register is set to 1 (multi-port sweep mode enabled), bits OPA0 and OPA 1 should be set to 00b.

Figure 18.3 AD0CON1 Register

A/D0 Control Register 2 (1)

b7	b6	b5	b4	b3	b2	b1	b0
0	0						

Symbol	Address	Reset Value
AD0CON2	0394h	XX0X X000b

Bit Symbol	Bit Name	Function	RW
SMP	A/D Conversion Method Select Bit	0: Without sample and hold function 1: With sample and hold function	RW
APS0	Analog Input Port Select Bit (2)	b2 b1 0 0 : AN_0 to AN_7, ANEX0, ANEX1 0 1 : AN15_0 to AN15_3 1 0 : AN0_0 to AN0_7 1 1 : AN2_0 to AN2_7	RW
APS1			RW
— (b4-b3)	No register bits; should be written with 0 and read as undefined value		—
TRG0	External Trigger Request Source Select Bit	0: Select $\overline{\text{ADTRG}}$ pin 1: Select a timer B2 interrupt request (after counting the ICTB2 register) in the three-phase motor control timers	RW
— (b7-b6)	Reserved	Should be written with 0 and read as undefined value	RW

Notes:
1. When this register is rewritten during an A/D conversion, the converted result is undefined.
2. Set bits APS1 and APS0 to 01b when the MSS bit in the AD0CON3 register is set to 1 (multi-port sweep mode enabled).

Figure 18.4 AD0CON2 Register

A/D0 Control Register 3 (1, 2)

b7	b6	b5	b4	b3	b2	b1	b0
0	0	0					

Symbol: AD0CON3 — Address: 0395h — Reset Value: XXXX X000b

Bit Symbol	Bit Name	Function	RW
DUS	DMAC Operating Mode Select Bit (3)	0: DMAC operating mode disabled 1: DMAC operating mode enabled (4, 5)	RW
MSS	Multi-port Sweep Mode Select Bit	0: Multi-port sweep mode disabled 1: Multi-port sweep mode enabled (3, 6)	RW
CKS2	Frequency Select Bit	(Note 7)	RW
MSF0	Multi-port Sweep Status Flag (8)	b4 b3 0 0 : AN_0 to AN_7 0 1 : AN15_0 to AN15_7 1 0 : AN0_0 to AN0_7 1 1 : AN2_0 to AN2_7	RO
MSF1			RO
— (b7-b5)	Reserved	Should be written with 0 and read as undefined value	RW

Notes:
1. When this register is rewritten during A/D conversion, the converted result is undefined.
2. This register may be read incorrectly during A/D conversion. It should be read or written after the A/D converter stops operating.
3. To set the MSS bit to 1, the DUS bit should be also set to 1.
4. When the DUS bit is set to 1, all A/D converted results are stored into the AD00 register.
5. To transfer converted results by DMA , configure the DMAC.
6. To set the MSS bit to 1, the following bit settings should be done:
   -the MD2 bit in the AD0CON1 register: 0 (mode other than repeat sweep mode 1)
   -bits APS1 and APS0 in the AD0CON2 register: 01b (AN15_0 to AN15_3)
   -bits OPA1 and OPA0 in the AD0CON1 register: 00b (no use of ANEX0 or ANEX1).
7. Refer to the note on the CKS0 bit in the AD0CON0 register.
8. This bit setting is enabled when the MSS bit is set to 1. The read value is undefined when the MSS bit is set to 0.

Figure 18.5 AD0CON3 Register

A/D0 Control Register 4 (1)

b7	b6	b5	b4	b3	b2	b1	b0
0	0	0	0			0	0

Symbol	Address	Reset Value
AD0CON4	0392h	XXXX 00XXb

Bit Symbol	Bit Name	Function	RW
— (b1-b0)	Reserved	Should be written with 0 and read as undefined value	RW
MPS10	Multi-port Sweep Port Select Bit	b3 b2 0 0 : (Note 2) 0 1 : AN_0 to AN_7, AN15_0 to AN15_7 (3) 1 0 : AN_0 to AN_7, AN0_0 to AN0_7 1 1 : AN_0 to AN_7, AN2_0 to AN2_7	RW
MPS11			RW
— (b7-b4)	Reserved	Should be written with 0 and read as undefined value	RW

Notes:
1. When this register is rewritten during A/D conversion, the converted result is undefined.
2. When the MSS bit in the AD0CON3 register is set to 0 (multi-port sweep mode disabled), set bits MSP11 and MPS10 to 00b. When it is set to 1 (multi-port sweep mode enabled), set them to any value other than 00b.
3. The converted result of signals AN15_4 to AN15_7 is undefined.

Figure 18.6 AD0CON4 Register

A/D0 Register i (i = 0 to 7) (1 to 4)

b15	b8	b7	b0

Symbol	Address	Reset Value
AD00, AD01	0381h-0380h, 0383h-0382h	0000 0000 XXXX XXXXb
AD02, AD03	0385h-0384h, 0387h-0386h	0000 0000 XXXX XXXXb
AD04, AD05	0389h-0388h, 038Bh-038Ah	0000 0000 XXXX XXXXb
AD06, AD07	038Dh-038Ch, 038Fh-038Eh	0000 0000 XXXX XXXXb

Bit Symbol	Function	RW
— (b7-b0)	The lower byte in an A/D converted result	RO
— (b9-b8)	In 10-bit mode: 2 upper bits in an A/D converted result In 8-bit mode: These bits are read as 0	RO
— (b15-b10)	These bits are read as 0	RO

Notes:
1. If this register is read by a program while the DMAC is configured to transfer converted results, the value is undefined.
2. Register value written while the A/D converter stops operating is undefined.
3. Only the AD00 register is available when the DUS bit in the AD0CON3 register is 1 (DMAC operating mode enabled). Other registers are undefined.
4. When a converted result is transferred by DMA at 10-bit mode, the DMAC should be set for a 16-bit transfer.

Figure 18.7 Registers AD00 to AD07

18.1 Mode Descriptions

18.1.1 One-shot Mode

In one-shot mode, the analog voltage applied to a selected pin is converted into a digital code only once. Table 18.2 lists specifications of one-shot mode.

Table 18.2 One-shot Mode Specification

Item	Specification
Function	Converts only once the analog voltage applied to a pin into a digital code. The pin is selected using bits CH2 to CH0 in the AD0CON0 register, OPA1 and OPA0 in the AD0CON1 register, and APS1 and APS0 in the AD0CON2 register
Start conditions	In the TRG bit setting in the AD0CON0 register to 0 (software trigger), the ADST bit in the AD0CON0 register is set to 1 (A/D conversion started) by a program. In the TRG bit setting to 1 (external trigger or hardware trigger), external trigger request source is selected by the TRG0 bit in the AD0CON2 register. • When 0 is selected, an input signal at the $\overline{\text{ADTRG}}$ pin switches from high to low after the ADST bit is set to 1 by a program. • When 1 is selected, generation of a timer B2 interrupt request which has passed through the circuit to set interrupt generating frequency in the three-phase motor control timers after the ADST bit is set to 1 by a program.
Stop conditions	• An A/D conversion is completed (the ADST bit is set to 0 when the software trigger is selected) • The ADST bit is set to 0 (A/D conversion stopped) by a program
Interrupt request generation timing	When the A/D conversion is completed, an interrupt request is generated
Input pin to be selected	One pin is selected from among AN_0 to AN_7, AN0_0 to AN0_7, AN2_0 to AN2_7, AN15_0 to AN15_3, ANEX0, and ANEX1
Reading of A/D converted result	In the DUS bit setting in the AD0CON3 register to 0 (DMAC operating mode disabled), read the AD0j register (j = 0 to 7) corresponding to the selected pin In the DUS bit setting to 1 (DMAC operating mode enabled), configure the DMAC (refer to 12. “DMAC”), then A/D converted result is stored in the AD00 register after the conversion is completed. The DMAC transfers the converted result from the AD00 register to given memory space. Do not read the AD00 register by a program

18.1.2 Repeat Mode

In repeat mode, the analog voltage applied to a selected pin is repeatedly converted into a digital code. Table 18.3 lists specifications of repeat mode.

Table 18.3 Repeat Mode Specification

Item	Specification
Function	Converts repeatedly the analog voltage input to a pin into a digital code. The pin is selected using bits CH2 to CH0 in the AD0CON0 register, OPA1 and OPA0 in the AD0CON1 register, and APS1 and APS0 in the AD0CON2 register
Start conditions	In the TRG bit setting in the AD0CON0 register to 0 (software trigger), the ADST bit in the AD0CON0 register is set to 1 (A/D conversion started) by a program. In the TRG bit setting to 1 (external trigger or hardware trigger), external trigger request source is selected by the TRG0 bit in the AD0CON2 register. • When 0 is selected, an input signal at the $\overline{\text{ADTRG}}$ pin switches from high to low after the ADST bit is set to 1 by a program. • When 1 is selected, generation of a timer B2 interrupt request which has passed through the circuit to set interrupt generating frequency in the three-phase motor control timers after the ADST bit is set to 1 by a program.
Stop conditions	• The ADST bit is set to 0 (A/D conversion stopped) by a program
Interrupt request generation timing	In the DUS bit setting in the AD0CON3 register to 0 (DMAC operating mode disabled), no interrupt request is generated. In the DUS bit setting to 1 (DMAC operating mode enabled), every time an A/D conversion is completed, an interrupt request is generated
Analog voltage input pins	One pin is selected from among AN_0 to AN_7, AN0_0 to AN0_7, AN2_0 to AN2_7, AN15_0 to AN15_3, ANEX0, and ANEX1
Reading of A/D converted result	In the DUS bit setting in the AD0CON3 register to 0 (DMAC operating mode disabled), read the AD0j register (j = 0 to 7) corresponding to the selected pin In the DUS bit setting to 1 (DMAC operating mode enabled), • when the converted result is transferred by DMA, configure the DMAC (refer to 12. “DMAC”), then A/D converted result is stored in the AD00 register after the conversion is completed. The DMAC transfers the converted result from the AD00 register to given memory space. Do not read the AD00 register by a program • when the converted result is transferred by a program, read the AD00 register after the IR bit in the AD0IC register becomes 1. Set the IR bit back to 0

18.1.3 Single Sweep Mode

In single sweep mode, the analog voltage applied to selected pins is converted one-by-one into a digital code. Table 18.4 lists specifications of single sweep mode.

Table 18.4 Single sweep mode Specification

Item	Specification
Function	Converts one-by-one the analog voltage input to a set of pins into a digital code. The pins are selected using bits SCAN1 and SCAN0 in the AD0CON1 register and APS1 and APS0 in the AD0CON2 register
Start conditions	In the TRG bit setting in the AD0CON0 register to 0 (software trigger), the ADST bit in the AD0CON0 register is set to 1 (A/D conversion started) by a program. In the TRG bit setting to 1 (external trigger or hardware trigger), external trigger request source is selected by the TRG0 bit in the AD0CON2 register. • When 0 is selected, an input signal at the $\overline{\text{ADTRG}}$ pin switches from high to low after the ADST bit is set to 1 by a program. • When 1 is selected, generation of a timer B2 interrupt request which has passed through the circuit to set interrupt generating frequency in the three-phase motor control timers after the ADST bit is set to 1 by a program.
Stop conditions	• An A/D conversion is completed (the ADST bit is set to 0 when the software trigger is selected) • The ADST bit is set to 0 (A/D conversion stopped) by a program
Interrupt request generation timing	In the DUS bit setting in the AD0CON3 register to 0 (DMAC operating mode disabled), when a sweep is completed, an interrupt request is generated. In the DUS bit setting to 1 (DMAC operating mode enabled), every time an A/D conversion is completed, an interrupt request is generated
Analog voltage input pins	Selected from a group of 2 pins (ANi_0 and ANi_1) (i = none, 0, 2, 15), 4 pins (ANi_0 to ANi_3), 6 pins (ANi_0 to ANi_5), or 8 pins (ANi_0 to ANi_7) (1)
Reading of A/D converted result	In the DUS bit setting in the AD0CON3 register to 0 (DMAC operating mode disabled), read the AD0j register (j = 0 to 7) corresponding to the selected pin In the DUS bit setting to 1 (DMAC operating mode enabled), configure the DMAC (refer to 12. “DMAC”), then A/D converted result is stored in the AD00 register after the conversion is completed. The DMAC transfers the converted result from the AD00 register to given memory space. Do not read the AD00 register by a program

Note:

1. When bits APS1 and APS0 are set to 01b (AN15_0 to AN15_3 selected), set bits SCAN1 and SCAN0 to 00b (ANi_0 and ANi_1 selected) or 01b (ANi_0 to ANi_3 selected).

18.1.4 Repeat Sweep Mode 0

In repeat sweep mode 0, the analog voltage applied to selected pins is repeatedly converted into a digital code. Table 18.5 lists specifications of repeat sweep mode 0.

Table 18.5 Repeat Sweep Mode 0 Specification

Item	Specification
Function	Converts repeatedly the analog voltage input to a set of pins into a digital code. The pins are selected using bits SCAN1 and SCAN0 in the AD0CON1 register and APS1 and APS0 in the AD0CON2 register
Start conditions	In the TRG bit setting in the AD0CON0 register to 0 (software trigger), the ADST bit in the AD0CON0 register is set to 1 (A/D conversion started) by a program. In the TRG bit setting to 1 (external trigger or hardware trigger), external trigger request source is selected by the TRG0 bit in the AD0CON2 register. • When 0 is selected, an input signal at the $\overline{\text{ADTRG}}$ pin switches from high to low after the ADST bit is set to 1 by a program. • When 1 is selected, generation of a timer B2 interrupt request which has passed through the circuit to set interrupt generating frequency in the three-phase motor control timers after the ADST bit is set to 1 by a program.
Stop conditions	• The ADST bit is set to 0 (A/D conversion stopped) by a program
Interrupt request generation timing	In the DUS bit setting in the AD0CON3 register to 0 (DMAC operating mode disabled), no interrupt request is generated. In the DUS bit setting to 1 (DMAC operating mode enabled), every time an A/D conversion is completed, an interrupt request is generated
Analog voltage input pins	Selected from a group of 2 pins (ANi_0 and ANi_1) (i = none, 0, 2, 15), 4 pins (ANi_0 to ANi_3), 6 pins (ANi_0 to ANi_5), or 8 pins (ANi_0 to ANi_7) (1)
Reading of A/D converted result	In the DUS bit setting in the AD0CON3 register to 0 (DMAC operating mode disabled), read the AD0j register (j = 0 to 7) corresponding to the selected pin In the DUS bit setting to 1 (DMAC operating mode enabled), • when the converted result is transferred by DMA, configure the DMAC (refer to 12. “DMAC”), then A/D converted result is stored in the AD00 register after the conversion is completed. The DMAC transfers the converted result from the AD00 register to given memory space. Do not read the AD00 register by a program • when the converted result is transferred by a program, read the AD00 register after the IR bit in the AD0IC register becomes 1. Set the IR bit back to 0

Note:
1. When bits APS1 and APS0 are set to 01b (AN15_0 to AN15_3 selected), set bits SCAN1 and SCAN0 to 00b (ANi_0 and ANi_1 selected) or 01b (ANi_0 to ANi_3 selected).

18.1.5 Repeat Sweep Mode 1

In repeat sweep mode 1, the analog voltage applied to eight selected pins including some prioritized pins is repeatedly converted into a digital code. Table 18.6 lists specifications of repeat sweep mode 1.

Table 18.6 Repeat Sweep Mode 1 Specification

Item	Specification
Function	Converts repeatedly the analog voltage input to a set of eight pins into a digital code. A/some selected pin(s) is/are converted by priority e.g. When AN_0 is prioritized, the analog voltage is converted into a digital code in the following order: AN_0→AN_1→AN_0→AN_2→AN_0→AN_3•••
	The eight pins are selected using bits SCAN1 and SCAN0 in the AD0CON1 register and APS1 and APS0 in the AD0CON2 register
Start conditions	In the TRG bit setting in the AD0CON0 register to 0 (software trigger), the ADST bit in the AD0CON0 register is set to 1 (A/D conversion started) by a program. In the TRG bit setting to 1 (external trigger or hardware trigger), external trigger request source is selected by the TRG0 bit in the AD0CON2 register. • When 0 is selected, an input signal at the $\overline{\text{ADTRG}}$ pin switches from high to low after the ADST bit is set to 1 by a program. Retrigger is invalid. • When 1 is selected, generation of a timer B2 interrupt request which has passed through the circuit to set interrupt generating frequency in the three-phase motor control timers after the ADST bit is set to 1 by a program.
Stop conditions	• The ADST bit is set to 0 (A/D conversion stopped) by a program
Interrupt request generation timing	In the DUS bit setting in the AD0CON3 register to 0 (DMAC operating mode disabled), no interrupt request is generated. In the DUS bit setting to 1 (DMAC operating mode enabled), every time an A/D conversion is completed, an interrupt request is generated
Analog voltage input pins	8 (ANi_0 to ANi_7) (i = none, 0, 2, 15) (1)
Prioritized pin(s)	Selected from a group of 1 pin (ANi_0), 2 pins (ANi_0 and ANi_1), 3 pins (ANi_0 to ANi_2), or 4 pins (ANi_0 to ANi_3)
Reading of A/D converted result	In the DUS bit setting in the AD0CON3 register to 0 (DMAC operating mode disabled), read the AD0j register (j = 0 to 7) corresponding to the selected pin In the DUS bit setting to 1 (DMAC operating mode enabled), • when the converted result is transferred by DMA, configure the DMAC (refer to 12. “DMAC”), then A/D converted result is stored in the AD00 register after the conversion is completed. The DMAC transfers the converted result from the AD00 register to given memory space. Do not read the AD00 register by a program • when the converted result is transferred by a program, read the AD00 register after the IR bit in the AD0IC register becomes 1. Set the IR bit back to 0

Note:
1. The converted result of signals AN15_4 to AN15_7 is undefined.

18.1.6 Multi-port Single Sweep Mode

In multi-port single sweep mode, the analog voltage applied to 16 selected pins is converted one-by-one into a digital code. The DUS bit in the AD0CON3 register should be set to 1 (DMAC operating mode enabled). Table 18.7 lists specifications of multi-port single sweep mode.

Table 18.7 Multi-port Single Sweep Mode Specification

Item	Specification
Function	Converts one-by-one the analog voltage input to a set of 16 selected pins into a digital code in the following order: AN_0 to AN_7→ANi_0 to ANi_7 (i = 0, 2, 15) The 16 pins are selected using bits MPS11 and MPS10 in the AD0CON4 register e.g. When bits MPS11 and MPS10 are set to 10b (AN_0 to AN_7, AN0_0 to AN0_7), the analog voltage is converted into a digital code in the following order: AN_0→AN_1→AN_2→AN_3→AN_4→AN_5→AN_6→AN_7→AN0_0→•••→AN0_6→AN0_7
Start conditions	In the TRG bit setting in the AD0CON0 register to 0 (software trigger), the ADST bit in the AD0CON0 register is set to 1 (A/D conversion started) by a program. In the TRG bit setting to 1 (external trigger or hardware trigger), external trigger request source is selected by the TRG0 bit in the AD0CON2 register. • When 0 is selected, an input signal at the $\overline{\text{ADTRG}}$ pin switches from high to low after the ADST bit is set to 1 by a program. • When 1 is selected, generation of a timer B2 interrupt request which has passed through the circuit to set interrupt generating frequency in the three-phase motor control timers after the ADST bit is set to 1 by a program.
Stop conditions	• An A/D conversion is completed (the ADST bit is set to 0 when the software trigger is selected) • The ADST bit is set to 0 (A/D conversion stopped) by a program
Interrupt request generation timing	Every time an A/D conversion is completed (Set the DUS bit to 1)
Analog voltage input pins	A combination of pin group is selected from AN_0 to AN_7→AN15_0 to AN15_7(1), AN_0 to AN_7→AN0_0 to AN0_7, or AN_0 to AN_7→AN2_0 to AN2_7
Reading of A/D converted result	Set the DUS bit to 1 and configure the DMAC (refer to 12. “DMAC”), then A/D converted result is stored in the AD00 register after the conversion is completed. The DMAC transfers the converted result from the AD00 register to given memory space. Do not read the AD00 register by a program

Note:
1. The converted result of signals AN15_4 to AN15_7 is undefined.

18.1.7 Multi-port Repeat Sweep Mode 0

In multi-port repeat sweep mode 0, the analog voltage applied to 16 selected pins is repeatedly converted into a digital code. The DUS bit in the AD0CON3 register should be set to 1 (DMAC operating mode enabled). Table 18.8 lists specifications of multi-port repeat sweep mode 0.

Table 18.8 Multi-port Repeat Sweep Mode 0 Specification

Item	Specification
Function	Converts repeatedly the analog voltage input to a set of 16 selected pins into a digital code in the following order: AN_0 to AN_7→ANi_0 to ANi_7 (i = 0, 2, 15) The 16 pins are selected using bits MPS11 and MPS10 in the AD0CON4 register e.g. When bits MPS11 and MPS10 are set to 10b (AN_0 to AN_7, AN0_0 to AN0_7), the analog voltage is repeatedly converted into a digital code in the following order: AN_0→AN_1→AN_2→AN_3→AN_4→AN_5→AN_6→AN_7→AN0_0→•••→AN0_6→AN0_7
Start conditions	In the TRG bit setting in the AD0CON0 register to 0 (software trigger), the ADST bit in the AD0CON0 register is set to 1 (A/D conversion started) by a program. In the TRG bit setting to 1 (external trigger or hardware trigger), external trigger request source is selected by the TRG0 bit in the AD0CON2 register. • When 0 is selected, an input signal at the $\overline{\text{ADTRG}}$ pin switches from high to low after the ADST bit is set to 1 by a program. • When 1 is selected, generation of a timer B2 interrupt request which has passed through the circuit to set interrupt generating frequency in the three-phase motor control timers after the ADST bit is set to 1 by a program.
Stop conditions	• The ADST bit is set to 0 (A/D conversion stopped) by a program
Interrupt request generation timing	Every time an A/D conversion is completed (Set the DUS bit to 1)
Analog voltage input pins	A combination of pin group is selected from AN_0 to AN_7→AN15_0 to AN15_7(1), AN_0 to AN_7→AN0_0 to AN0_7, or AN_0 to AN_7→AN2_0 to AN2_7
Reading of A/D converted result	Set the DUS bit to 1 and configure the DMAC (refer to 12. “DMAC”), then A/D converted result is stored in the AD00 register after the conversion is completed. The DMAC transfers the converted result from the AD00 register to given memory space. Do not read the AD00 register by a program

Note:
1. The converted result of pins AN15_4 to AN15_7 is undefined.

18.2 Functions

18.2.1 Resolution Selection

The resolution is selected using the BITS bit in the AD0CON1 register. When the BITS bit is set to 1 (10-bit precision), the A/D converted result is stored into bits 9 to 0 in the AD0i register (i = 0 to 7). When the BITS bit is set to 0 (8-bit precision), the result is stored into bits 7 to 0 in the AD0i register.

18.2.2 Sample and Hold Function

This function improves the conversion rate per pin to 28 ϕAD cycles at 8-bit resolution and 33 ϕAD cycles for 10-bit resolution. To use this function, which is available in all operating modes, set the SMP bit in the AD0CON2 register to 1 (with sample and hold function). Start the A/D conversion after setting the SMP bit.

18.2.3 Trigger Selection

A trigger to start A/D conversion is specified by the combination of TRG bit in the AD0CON0 register and the TRG0 bit in the AD0CON2 register. Table 18.9 lists the settings of the trigger selection.

Table 18.9 Trigger Selection Settings

Bit and Setting		Trigger
AD0CON0 register	AD0CON2 register	
TRG = 0	—	Software trigger The ADST bit in the AD0CON0 register is set to 1
TRG = 1 (1, 2)	TRG0 = 0	External trigger Falling edge of a signal applied to the $\overline{\text{ADTRG}}$ pin
	TRG0 = 1	Hardware trigger Generation of a timer B2 interrupt request which has passed through the circuit to set interrupt generating frequency in the three-phase motor control timers

Notes:
1. The A/D converter starts operating if a trigger is generated while the ADST bit is set to 1 (A/D conversion started).
2. If an external trigger or a hardware trigger is generated during an A/D conversion, the A/D converter aborts the operation in progress. Then, it resumes the operation.

18.2.4 DMAC Operating Mode

The DMAC operating mode is available in all operating modes. When the A/D converter is in multi-port single sweep mode or multi-port repeat sweep mode 0, the DMAC operating mode should be used definitely. When the DUS bit in the AD0CON3 register is set to 1 (DMAC operating mode enabled), all A/D converted results are stored into the AD00 register. The DMAC transfers the data from the AD00 register to a given memory space every time an A/D conversion is completed at a pin. 8-bit DMA transfer should be selected for 8-bit resolution. For 10-bit resolution, 16-bit DMA transfer should be selected. Refer to 12. “DMAC” for details.

18.2.5 Function-extended Analog Input Pins

In one-shot mode and repeat mode, pins ANEX0 and ANEX1 are available as analog input by setting bits OPA1 and OPA0 in the AD0CON1 register (refer to Table 18.10). The A/D converted result of pins ANEX0 and ANEX1 are respectively stored into registers AD00 and AD01. However, when the DUS bit in the AD0CON3 register is set to 1 (DMAC operating mode enabled), all results are stored into the AD00 register.

To use function-extended analog input pins, bits APS1 and APS0 in the AD0CON2 register should be set to 00b (AN0 to AN7, ANEX0, ANEX1 as analog input port) and the MSS bit in the AD0CON3 register to 0 (multi-port sweep mode disabled).

Table 18.10 Function-extended Analog Input Pin Settings

AD0CON1 Register		ANEX0	ANEX1
OPA1	OPA0		
0	0	Not used	Not used
0	1	Analog input	Not used
1	0	Not used	Analog input
1	1	Output to an external op-amp	Input from an external op-amp

18.2.6 External Operating Amplifier (Op-AMP) Connection Mode

In external op-amp connection mode, multiple analog inputs can be amplified by one external op-amp using function-extended analog input pins ANEX0 and ANEX1.

When bits OPA1 and OPA0 in the AD0CON1 register are set to 11b (external op-amp connected), the voltage applied to pins AN0 to AN7 are output from the ANEX0 pin. This output signal should be amplified by an external op-amp and applied to the ANEX1 pin.

The analog voltage applied to the ANEX1 pin is converted into a digital code. The converted result is stored into the corresponding AD0i register (i = 0 to 7). The conversion rate varies with the response of the external op-amp. The ANEX0 pin should not be connected to the ANEX1 pin directly.

To use external op-amp connection mode, bits APS1 and APS0 in the AD0CON2 register should be set to 00b (AN0 to AN7, ANEX0, ANEX1 as analog input port).

Figure 18.8 shows an example of an external op-amp connection.

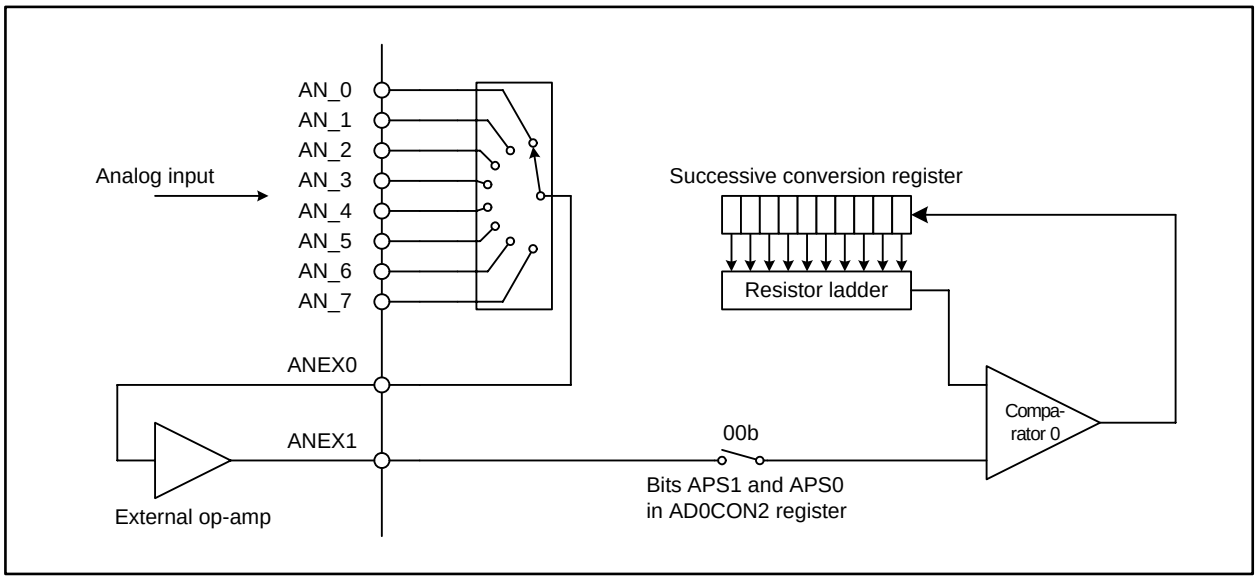


Figure 18.8 External Op-Amp Connection

18.2.7 Power Saving

When the A/D converter is not in use, power consumption can be reduced by setting the VCUT bit in the AD0CON1 to 0 (VREF disconnected). With this bit setting the reference voltage input pin (VREF) can be disconnected from the resistor ladder, which enables the power supply from the VREF to the resistor ladder to stop.

To use the A/D converter, the VCUT bit should be set to 1 (VREF connected) and 1 µs or more after, the ADST bit in the AD0CON0 register should be set to 1 (A/D conversion started). Bits ADST and VCUT should not be set to 1 simultaneously. The VCUT bit should not be set to 0 during the A/D conversion. The VCUT bit does not affect the VREF performance of the D/A converter (Refer to Figure 18.9).

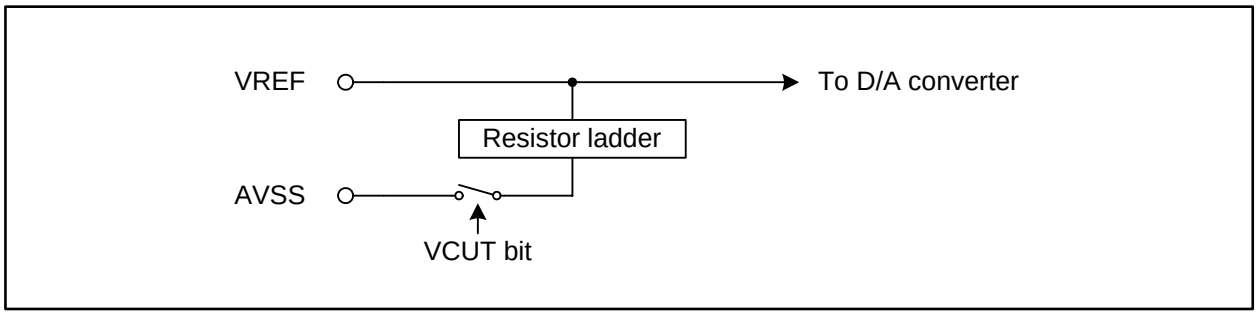


Figure 18.9 Power Supply by VCUT Bit

18.2.8 Output Impedance of Sensor Equivalent Circuit under A/D Conversion

Figure 18.10 shows an analog input pin and external sensor equivalent circuit.

To perform A/D conversion correctly, internal capacitor (C) charging, shown in Figure 18.10, should be completed within the specified period. This period, called sampling time, is 2 ϕAD cycles for conversion without the sample and hold function and 3 ϕAD cycles for conversion with this function.

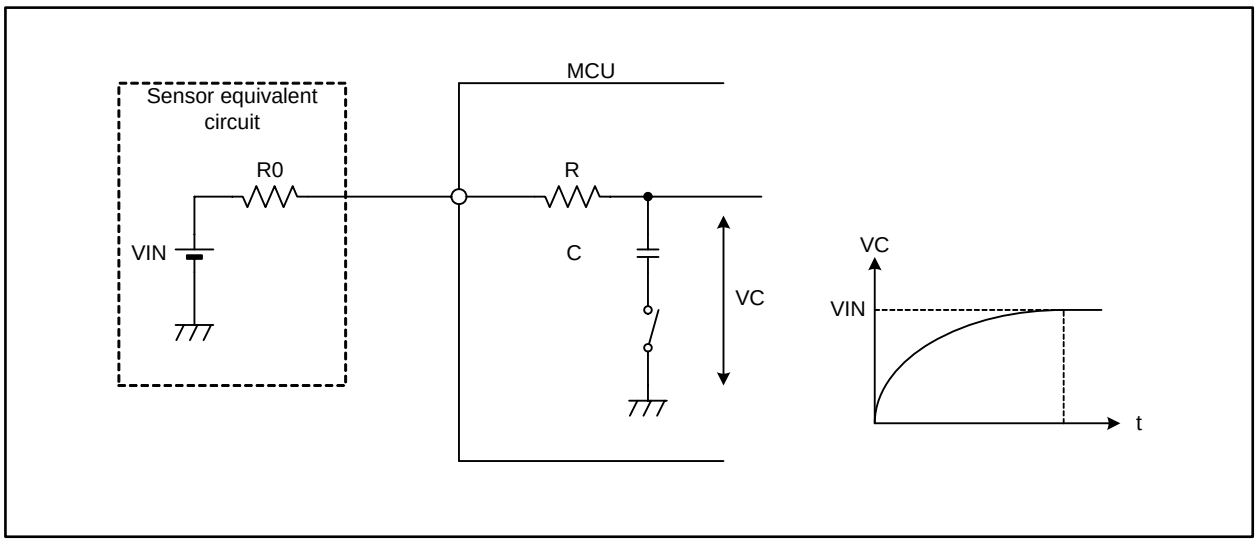


Figure 18.10 Analog Input Pin and External Sensor Equivalent Circuitry

The voltage between pins (VC) is expressed as follows:

$$VC = VIN\left\{1 - e^{-\frac{t}{C(R0+R)}}\right\}$$

When t = T and the precision (error) is x or less,

$$VC = VIN - \frac{x}{y}VIN = VIN\left(1 - \frac{x}{y}\right)$$

Thus, output impedance of the sensor equivalent circuit (R0) is determined by the following formulas:

$$e^{-\frac{T}{C(R0+R)}} = \frac{x}{y}$$

$$-\frac{T}{C(R0+R)} = \ln\frac{x}{y}$$

$$R0 = -\frac{T}{C\ln\frac{x}{y}} - R$$

where:
T[s] = Sampling time
R0[Ω] = Output Impedance of the sensor equivalent circuit
VC = Potential difference between edges of the capacitor C
R[Ω] = Internal resistance of the MCU
x[LSB] = Precision (error) of the A/D converter
y[step] = Resolution of the A/D converter (1024 steps @10-bit mode, 256 steps @ 8-bit mode)

When ϕAD = 10 MHz, the A/D conversion mode is 10-bit resolution with the sample and hold function, the output impedance (R0) with the precision (error) of 0.1 LSB or less is determined by the following formula:
Using T = 0.3 µs, R = 2.0 kΩ (reference value), C = 6.5 pF (reference value), x = 0.1, y = 1024,

$$R0 = -\frac{0.3\times10^{-6}}{6.5\times10^{-12}\times1\text{n}\frac{0.1}{1024}} - 2.0\times10^{3}$$

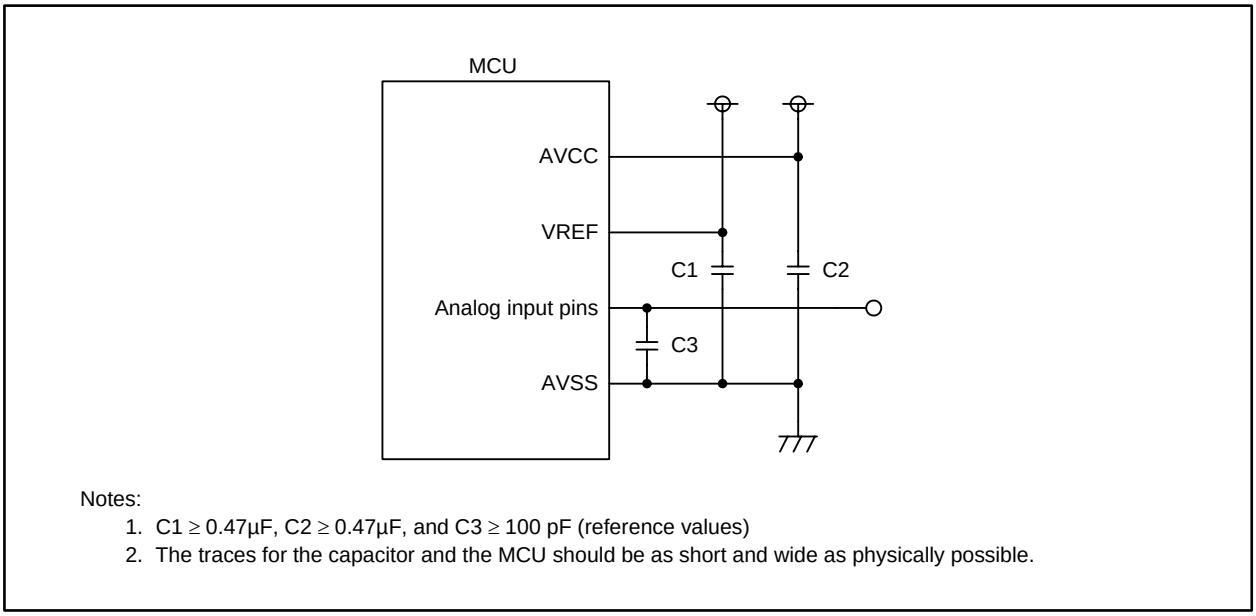
$$= 2998$$

Thus, the allowable output impedance of the sensor equivalent circuit (R0), making the precision (error) of 0.1 LSB or less, should be less than 3 kΩ.
Actual error, however, is the value of absolute precision added to 0.1 LSB mentioned above.

18.3 Notes on A/D Converter

18.3.1 Notes on Designing Boards

• Three capacitors should be respectively placed between the AVSS pin and such pins as AVCC, VREF, and analog inputs (AN_0 to AN_7, AN0_0 to AN0_7, AN2_0 to AN2_7, and AN15_0 to AN15_3) to avoid error operations caused by noise or latchup, and to reduce conversion errors. Figure 18.11 shows an example of pin configuration for A/D converter.



Notes:
1. C1 ≥ 0.47µF, C2 ≥ 0.47µF, and C3 ≥ 100 pF (reference values)
2. The traces for the capacitor and the MCU should be as short and wide as physically possible.

Figure 18.11 Pin Configuration for A/D Converter

• Do not use any of the four pins AN_4 to AN_7 for analog input if the key input interrupt is to be used. Otherwise, a key input interrupt request occurs when the A/D input voltage becomes VIL or lower.

• When AVCC = VREF = VCC, A/D input voltage for pins AN_0 to AN_7, AN0_0 to AN0_7, AN2_0 to AN2_7, AN15_0 to AN15_3, ANEX0, and ANEX1 should be VCC or lower.

18.3.2 Notes on Programming

- The following registers should be written while the A/D conversion is stopped, that is, before a trigger occurs: AD0CON0 (except the ADST bit), AD0CON1, AD0CON2, AD0CON3, and AD0CON4.

- If the VCUT bit in the AD0CON1 register is switched from 0 (VREF connected) to 1 (VREF disconnected), the A/D conversion should be started after 1 µs or more. Set the VCUT bit to 0 when A/D conversion is not used to reduce power consumption.

- Set the port direction bit for the pin to be used as an analog input pin to 0 (input). Set the ASEL bit of the corresponding port function select register to 1 (the port is used as A/D input).

- If the TRG bit in the AD0CON0 register is set to 1 (external trigger or hardware trigger is selected), set the corresponding port direction bit (PD9_7 bit) for the $\overline{\text{ADTRG}}$ pin to 0 (input).

- The ϕAD frequency should be 10 MHz or below. It should be 1 MHz or above if the sample and hold function is enabled. If not, it should be 250 kHz or above.

- If A/D operating mode (bits MD1 and MD0 in the AD0CON0 register or the MD2 bit in the AD0CON1 register) has been changed, re-select analog input pins by using bits CH2 to CH0 in the AD0CON0 register or bits SCAN1 and SCAN0 in the AD0CON1 register.

- If the AD0i register (i = 0 to 7) is read when the A/D converted result is stored to the register, the stored value may have an error. Read the AD0i register after the A/D conversion has been completed.
  In one-shot mode or single sweep mode, read the respective AD0i register after the IR bit in the AD0IC register has become 1 (interrupt requested).
  In repeat mode, repeat sweep mode 0, or repeat sweep mode 1, an interrupt request can be generated each time when an A/D conversion has been completed if the DUS bit in the AD0CON3 register is set to 1 (DMAC operating mode enabled). Similar to the other modes above, read the AD00 register after the IR bit in the AD0IC register has become 1 (interrupt requested).

- If the A/D conversion in progress is halted by setting the ADST bit in the AD0CON0 register to 0, the converted result is undefined. In addition, the unconverted AD0i register may also become undefined. Consequently, the AD0i register should not be used just after A/D conversion is halted.

- The external trigger cannot be used in DMAC operating mode. When the DMAC is configured to transfer converted results, do not read the AD00 register by a program.

- If, in single sweep mode, the A/D conversion in progress is halted by setting the ADST bit in the AD0CON0 register to 0 (A/D conversion is stopped), an interrupt request may be generated even though the sweep is not completed. To halt the A/D conversion, first disable interrupts, then set the ADST bit to 0.

19. D/A Converter

The MCU has two separate 8-bit R-2R resistor ladder D/A converters.
Digital code is converted to an analog voltage when a value is written to the corresponding DAi register (i = 0,1). The DAiE bit in the DACON register determines whether the D/A conversion result is output or not. To output the converted value, the DAiE bit should be set to 1 (output enabled). This bit setting disables a pull-up resistor for the corresponding port.
Analog voltage to be output (V) is calculated based on the value (n) set in the DAi register (n = decimal).

$$V = \frac{VREF \times n}{256} \qquad \text{(n = 0 to 255)}$$

VREF: reference voltage

Table 19.1 lists specifications of the D/A converter. Figure 19.1 shows a block diagram of the D/A converter. Figure 19.2 and Figure 19.3 show registers associated with the D/A. Figure 19.4 shows a D/A converter equivalent circuit.
When the D/A converter is not used, the DAi register should be set to 00h and the DAiE bit should be set to 0 (output disabled).

Table 19.1 D/A Converter Specifications

Item	Specification
D/A conversion method	R-2R resistor ladder
Resolution	8 bits
Analog output pins	2 channels

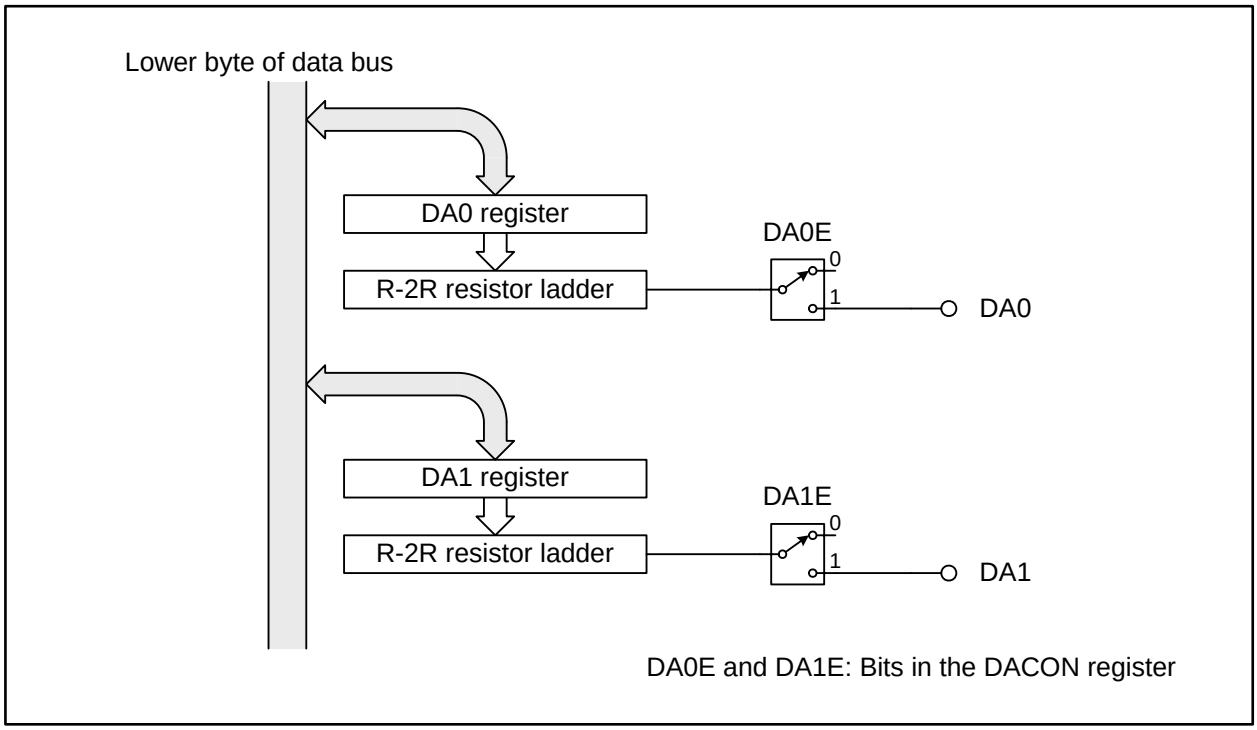


Figure 19.1 D/A Converter Block Diagram

D/A Control Register

b7 b6 b5 b4 b3 b2 b1 b0

Symbol: DACON
Address: 039Ch
Reset Value: XXXX XX00b

Bit Symbol	Bit Name	Function	RW
DA0E	D/A0 Output Enable Bit	0: Output disabled 1: Output enabled	RW
DA1E	D/A1 Output Enable Bit	0: Output disabled 1: Output enabled	RW
— (b7-b2)	No register bits; should be written with 0 and read as undefined value		—

Figure 19.2 DACON Register

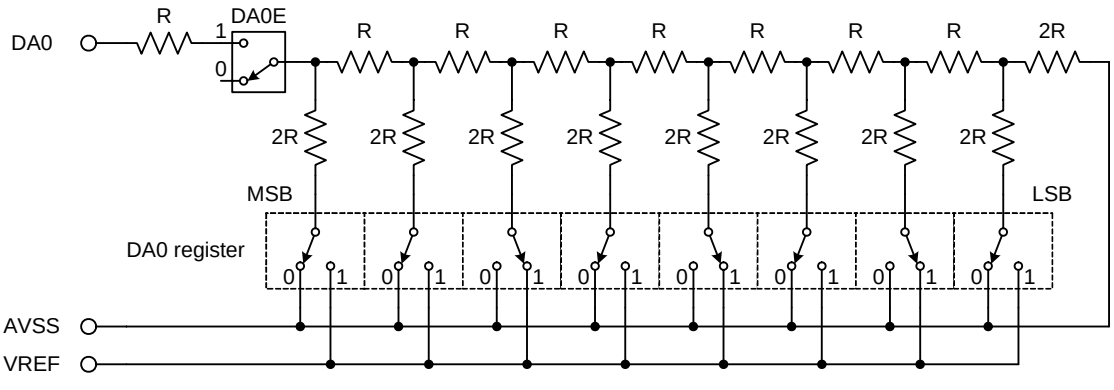
D/A Register i (i = 0, 1)

b7 b0

Symbol: DA0, DA1
Address: 0398h, 039Ah
Reset Value: Undefined

Function	Setting Range	RW
Output value by the D/A conversion	00h to FFh	RW

Figure 19.3 Registers DA0 and DA1



Notes:
1. The figure above applies when the DA0 register is set to 2Ah.
2. This circuitry also applies to the D/A converter 1.
3. To reduce power consumption when the D/A converter is not in use, the DAiE bit (i = 0, 1) should be set to 0 (output disabled) and the DAi register should be set to 00h to prevent the current from flowing into the R-2R resistor ladder.

Figure 19.4 D/A Converter Equivalent Circuitry

20. CRC Calculator

The CRC (Cyclic Redundancy Check) calculator is used for error detection in data blocks. A generator polynomial of CRC-CCITT ($X^{16} + X^{12} + X^5 + 1$) generates a CRC.
The CRC is a 16-bit code generated for given blocks of 8-bit data. It is set in the CRCD register every time 1-byte data is written to the CRCIN register after a default value is set to the CRCD register.
Figure 20.1 shows a block diagram of the CRC calculator. Figure 20.2 and Figure 20.3 show registers associated with the CRC. Figure 20.4 shows an example of the CRC calculation.

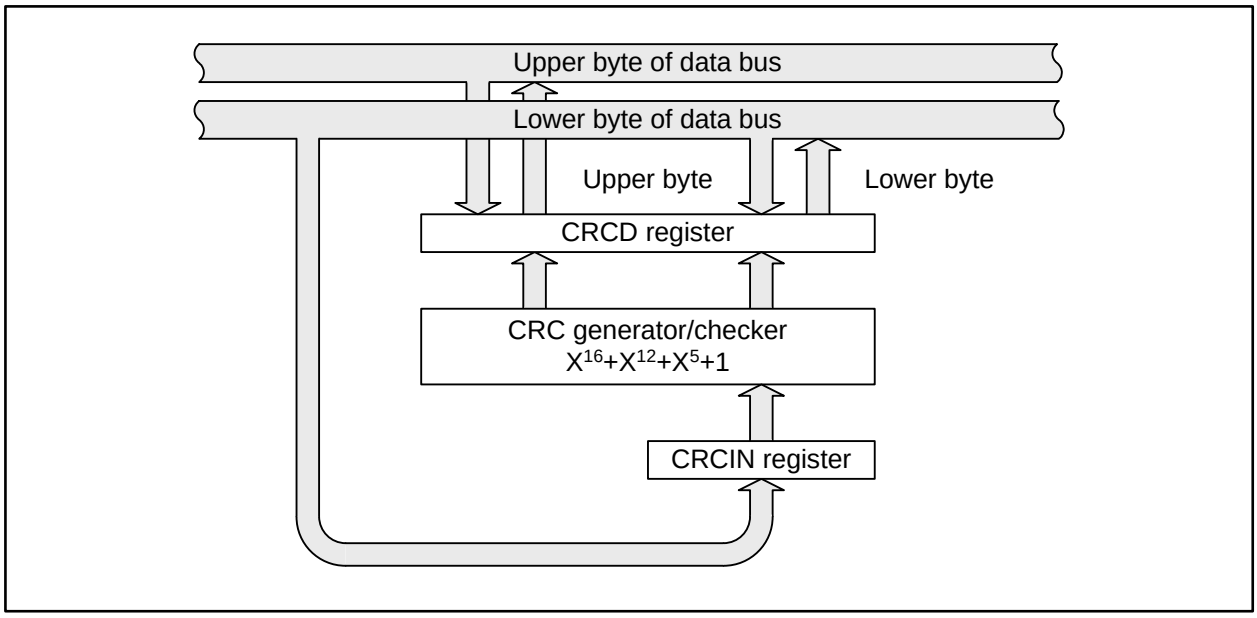


Figure 20.1 CRC Calculator Block Diagram

CRC Data Register

b15 b8 b7 b0

Symbol	Address	Reset Value
CRCD	037Dh-037Ch	Undefined

Function	Setting Range	RW
The CRC calculation result is stored in the CRCD register. When a default value in reversed bit position is set in this register and then data in reversed bit position is written to the CRCIN register, the CRC in reversed bit position is read from this register	0000h to FFFFh	RW

Figure 20.2 CRCD Register

CRC Input Register

b7 b0

Symbol	Address	Reset Value
CRCIN	037Eh	Undefined

Function	Setting Range	RW
This register is to input data. Data to be input should be in reversed bit position	00h to FFh	RW

Figure 20.3 CRCIN Register

CRC Calculation and Setting Procedure to Generate CRC for 80C4h

- CRC Calculation for R32C

  CRC: a remainder of the division as follows: $\frac{\text{reversed-bit-position value in the CRCIN register}}{\text{generator polynominal}}$

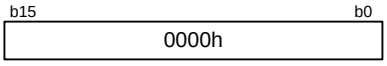
  Generator Polynomial: $X^{16}+X^{12}+X^5+1$(1 0001 0000 0010 0001b)

- Setting Steps

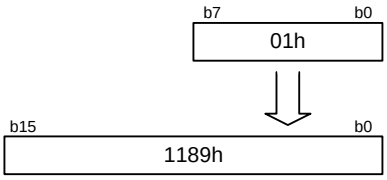
  (1) Reverse the bit position of 80C4h per byte by a program

  80h to 01h, C4h to 23h

  (2) Set 0000h (default value in reversed bit position) in CRCD register

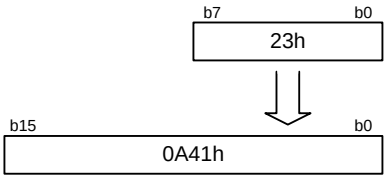
  

  (3) Set 01h (80h in reversed bit position) in CRCIN register

  1189h, CRC for 80h (9188h) in reversed bit position is stored into the CRCD register in the third cycle.

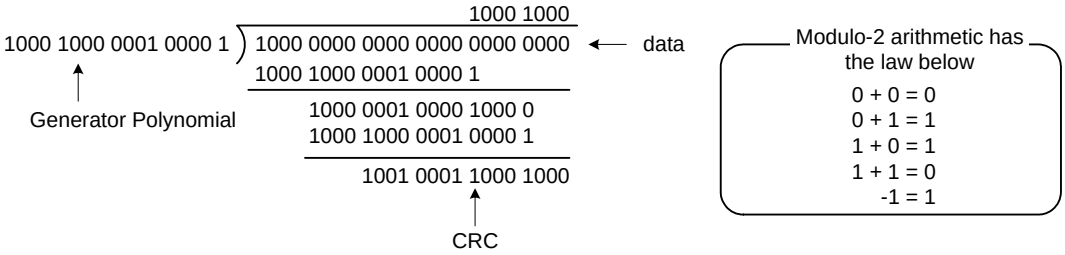
  (4) Set 23h (C4h in reversed bit position) in CRCIN register

  0A41h, CRC for 80C4h (8250h) in revered bit position is stored into the CRCD register in the third cycle.

- Details of CRC Calculation

As shown in (3) above, add 1000 0000 0000 0000 0000 0000b as 80h (1000 0000b) plus 16 digits to 0000 0000 0000 0000 0000 0000b as the default value of the CRCD register, 0000h plus eight digits to perform the modulo-2 division.



0001 0001 1000 1001b (1189h), the reversed-bit-position value of remainder 1001 0001 1000 1000b (9188h) can be read from the CRCD register.

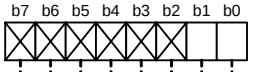
When going on (4) above, add 1100 0100 0000 0000 0000 0000b as C4h (1100 0100b) plus 16 digits to 1001 0001 1000 1000 0000 0000b as the remainder of (3) left in the CRCD register plus eight digits to perform the modulo-2 division.
0000 1010 0100 0001b (0A41h), the reversed-bit-position value of remainder 1000 0010 0101 0000b (8250h) can be read from the CRCD register.

Figure 20.4 CRC Calculation

21. X-Y Conversion

The X-Y conversion rotates a 16 × 16-bit matrix data 90 degrees or reverses the bit position of 16-bit data. The X-Y conversion is set using the XYC register shown in Figure 21.1.
Data is written in write-only XiR registers (i = 0 to 15) and converted data is read in read-only YjR register (j = 0 to 15). These registers are allocated to the same address. Figure 21.2 and Figure 21.3 show registers XiR and YjR, respectively. A write/read access from an even address to the XiR/YjR registers should be performed every 16 bits. 8-bit access operation results are undefined.

X-Y Control Register

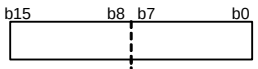


Symbol	Address	Reset Value
XYC	02E0h	XXXX XX00b

Bit Symbol	Bit Name	Function	RW
XYC0	Read Mode Set Bit	0: Data rotation 1: No data rotation	RW
XYC1	Write Mode Set Bit	0: No bit position reverse 1: Bit position reverse	RW
— (b7-b2)	No register bits; should be written with 0 and read as undefined value		—

Figure 21.1 XYC Register

Xi Register (i = 0 to 15) (1)



Symbol	Address	Reset Value
X0R to X2R	02C1h-02C0h, 02C3h-02C2h, 02C5h-02C4h	Undefined
X3R to X5R	02C7h-02C6h, 02C9h-02C8h, 02CBh-02CAh	Undefined
X6R to X8R	02CDh-02CCh, 02CFh-02CEh, 02D1h-02D0h	Undefined
X9R to X11R	02D3h-02D2h, 02D5h-02D4h, 02D7h-02D6h	Undefined
X12R to X14R	02D9h-02D8h, 02DBh-02DAh, 02DDh-02DCh	Undefined
X15R	02DFh-02DEh	Undefined

Function	Setting Range	RW
Input data for X-Y conversion	0000h to FFFFh	WO

Note:
1. A 16-bit write access to this register should be performed.

Figure 21.2 Registers X0R to X15R

Yj Register (j = 0 to 15) (1)

b15 b8 b7 b0

Symbol	Address	Reset Value
Y0R to Y2R	02C1h-02C0h, 02C3h-02C2h, 02C5h- 02C4h	Undefined
Y3R to Y5R	02C7h-02C6h, 02C9h-02C8h, 02CBh-02CAh	Undefined
Y6R to Y8R	02CDh-02CCh, 02CFh-02CEh, 02D1h-02D0h	Undefined
Y9R to Y11R	02D3h-02D2h, 02D5h-02D4h, 02D7h-02D6h	Undefined
Y12R to Y14R	02D9h-02D8h, 02DBh-02DAh, 02DDh-02DCh	Undefined
Y15R	02DFh-02DEh	Undefined

Function	RW
Result of X-Y conversion	RO

Note:
1. A 16-bit read access to this register should be performed.

Figure 21.3 Registers Y0R to Y15R

21.1 Data Conversion on Reading

The XYC0 bit in the XYC register selects a read mode for the YjR register. When the XYC0 bit is set to 0 (data rotation), bit j in the corresponding registers X0R to X15R is automatically read on reading the YjR register (j = 0 to 15).

More concretely, on reading bit i (i = 0 to 15) in the Y0R register, the data of each bit 0 in the XiR register is read. That is, a read data of bit 0 in the Y15R register means the data of bit 15 in the X0R register and the data of bit 15 in the Y0R register is identical to that of bit 0 in the X15R register.

Figure 21.4 shows the conversion table when the XYC0 bit is set to 0 and Figure 21.5 shows an example of X-Y conversion.

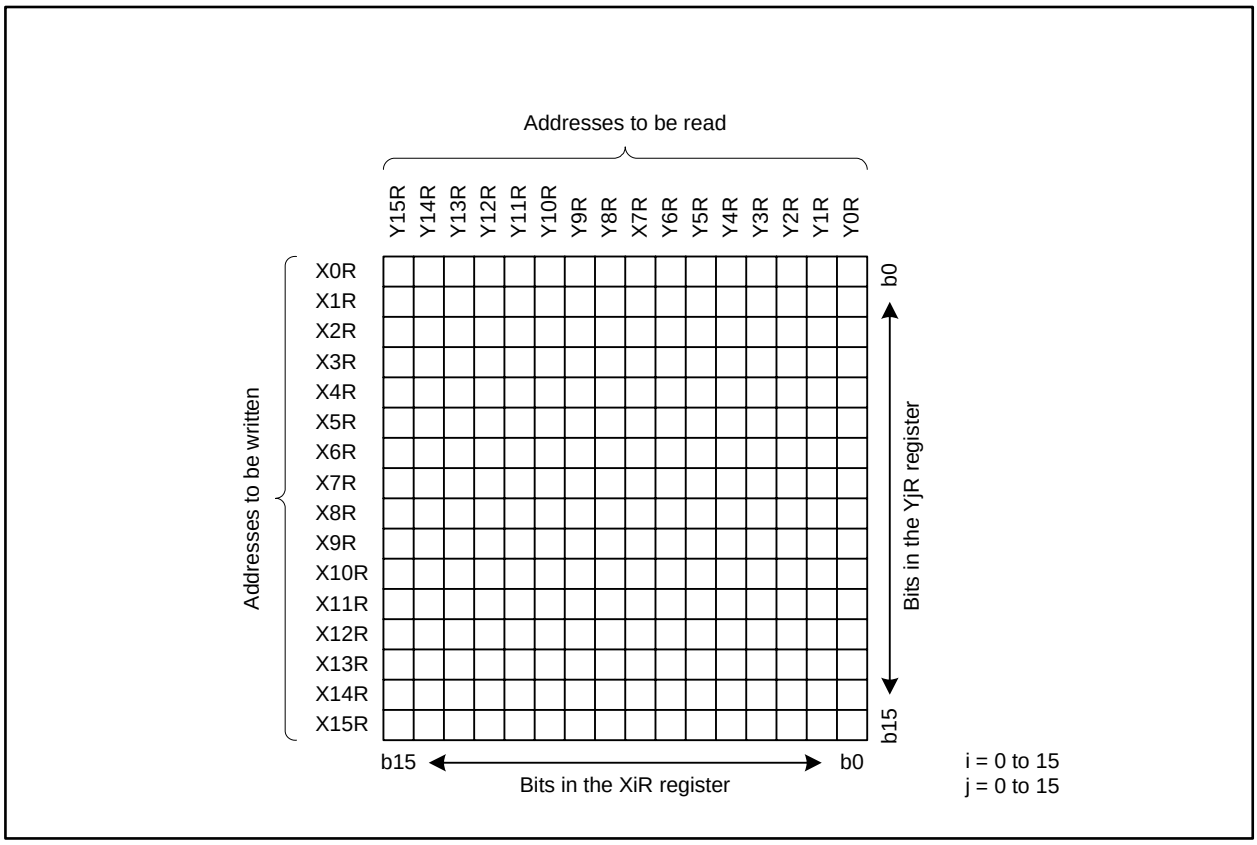


Figure 21.4 Conversion Table (XYC0 Bit = 0)

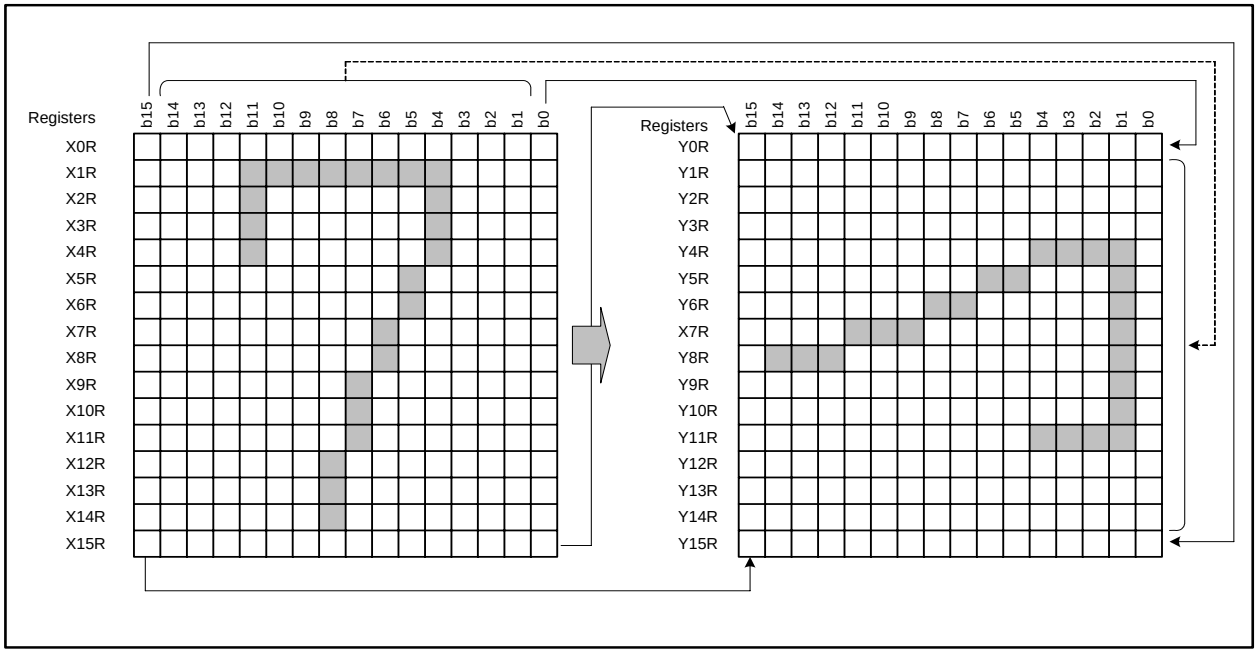


Figure 21.5 X-Y Conversion

When the XYC0 bit is set to 1 (no data rotation), the data of each bit in the YjR register is identical to that written in the XiR register. Figure 21.6 shows the conversion table when the XYC0 bit is set to 1.

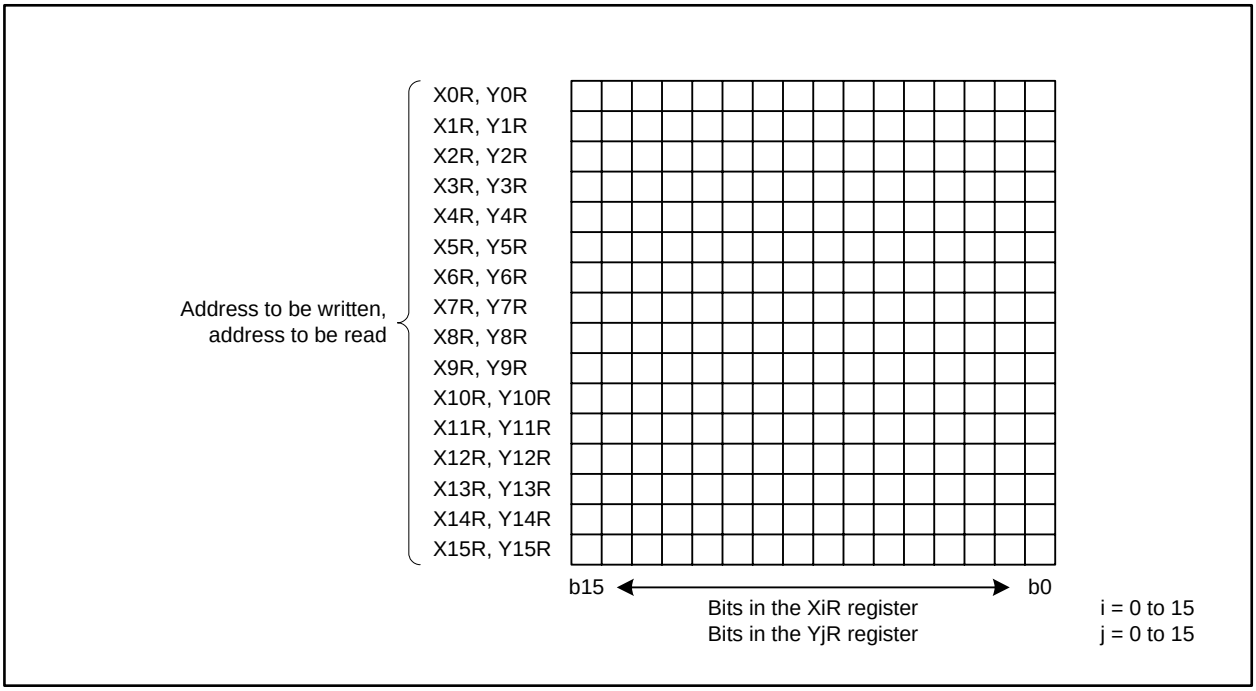


Figure 21.6 Conversion Table (XYC0 Bit = 1)

21.2 Data Conversion on Writing

The XYC1 bit in the XYC register selects a write mode for the XiR register.
When the XYC1 bit is set to 0 (no bit position reverse), the data is written in order. When it is set to 1 (bit position reverse), the data is written in reversed order. Figure 21.7 shows the conversion table when the XYC1 bit is set to 1.

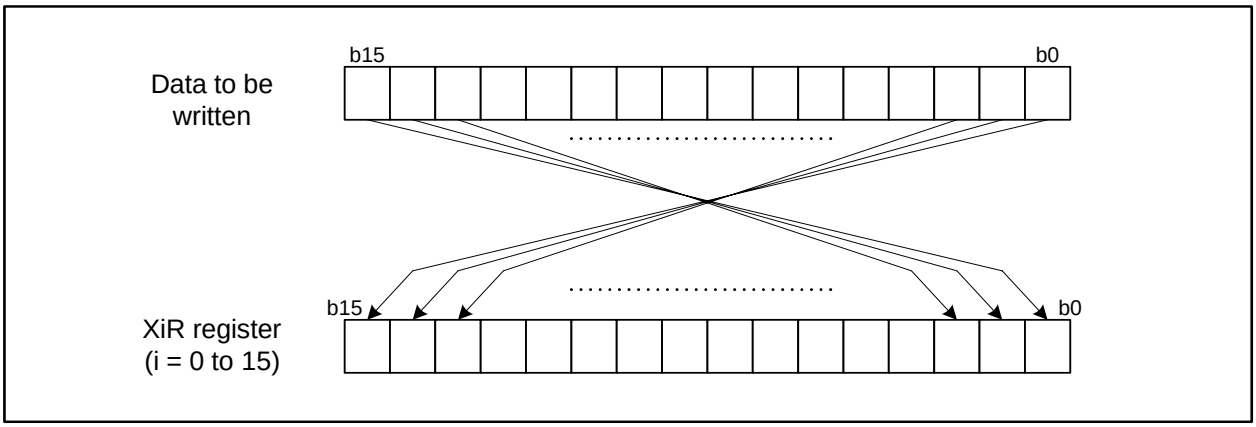


Figure 21.7 Conversion Table (XYC1 Bit = 1)

22. Intelligent I/O

The intelligent I/O is a multifunctional I/O port for time measurement, waveform generation, variable character length synchronous serial interface, and IEBus.
It consists of three groups each of which has one free-running 16-bit base timer and eight 16-bit registers for time measurement or waveform generation.
Table 22.1 lists functions and channels of the intelligent I/O.

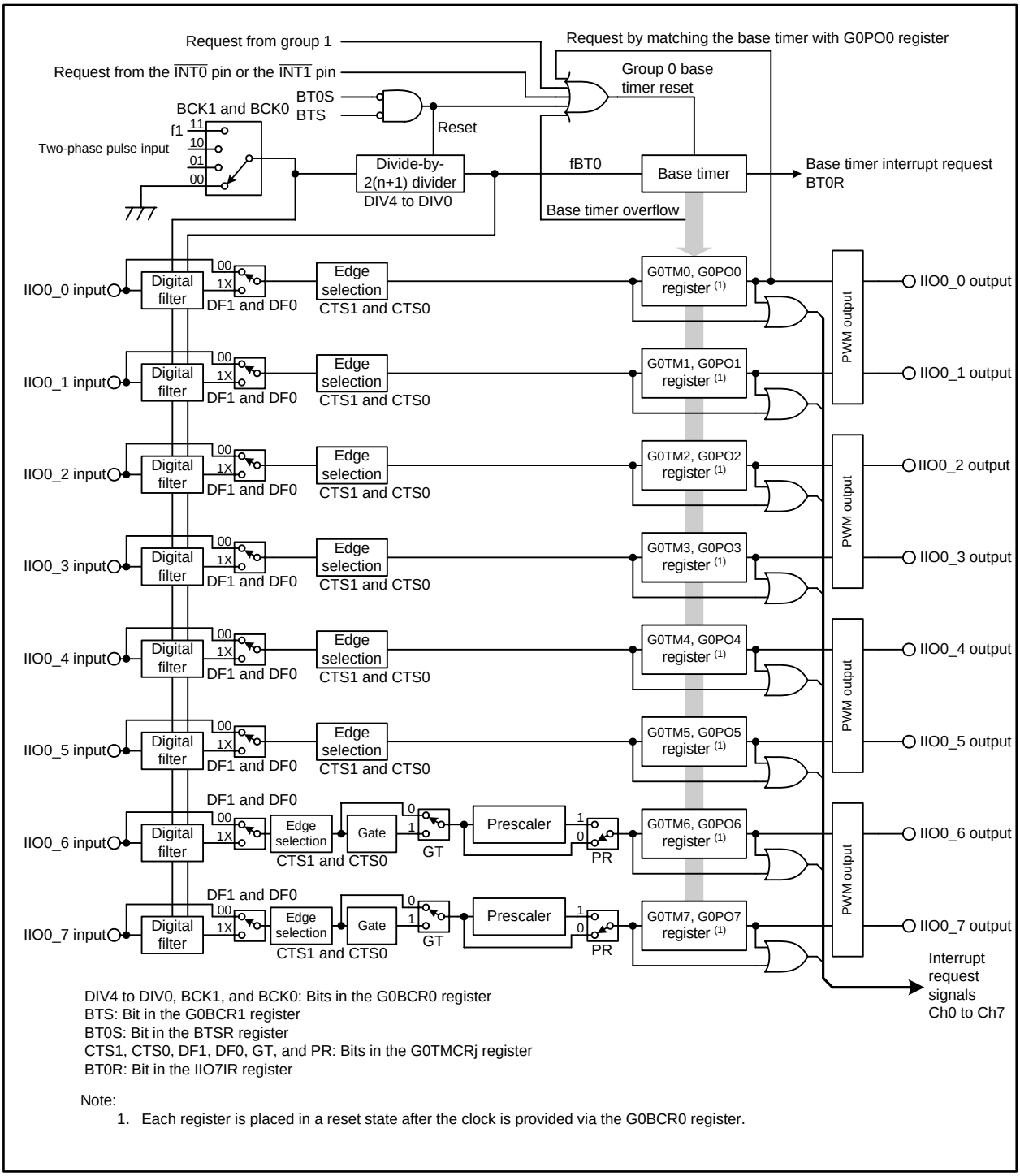
Table 22.1 Intelligent I/O Functions and Channels

Functions		Group 0	Group 1	Group 2
Time measurement (1)	Digital filter	8 channels	8 channels	Not available
	Prescaler	2 channels	2 channels	
	Gating	2 channels	2 channels	
Waveform generation (1)	Single-phase waveform output mode	8 channels	8 channels	8 channels
	Inverted waveform output mode	8 channels	8 channels	8 channels
	SR waveform output mode	8 channels	8 channels	8 channels
	Bit modulation PWM mode	Not available	Not available	8 channels
	RTP mode			8 channels
	Parallel RTP mode			8 channels
Serial interface	Variable character length synchronous serial interface mode	Not available	Not available	Available
	IEBus mode (optional (2))			

Notes:
1. Functions time measurement and waveform generation share a pin.
2. Contact a Renesas Electronics sales office to use the optional features.

Each channel individually selects a function from the time measurement and the waveform generation.

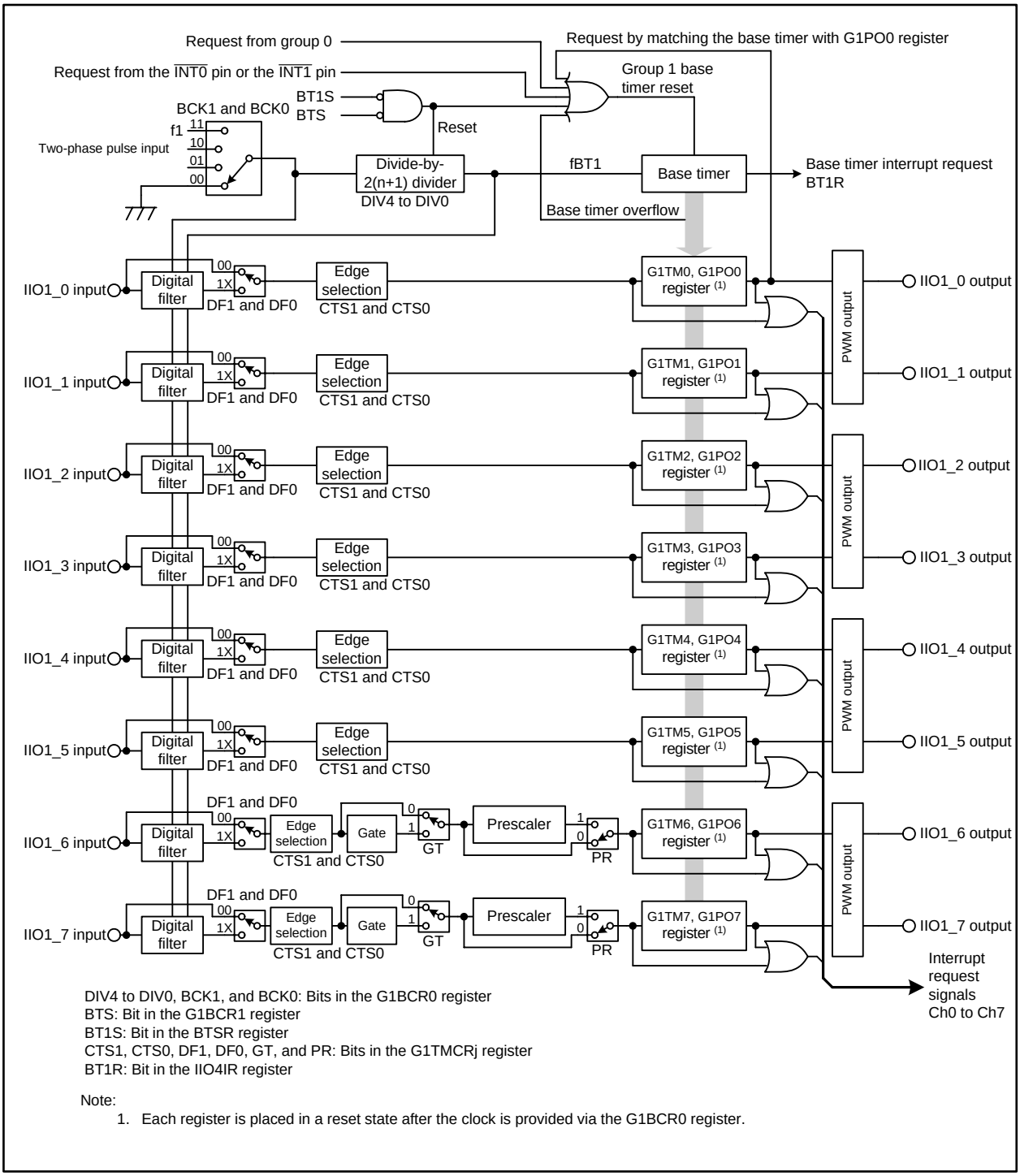
Figure 22.1 to Figure 22.3 show block diagrams of the intelligent I/O.



DIV4 to DIV0, BCK1, and BCK0: Bits in the G0BCR0 register
BTS: Bit in the G0BCR1 register
BT0S: Bit in the BTSR register
CTS1, CTS0, DF1, DF0, GT, and PR: Bits in the G0TMCRj register
BT0R: Bit in the IIO7IR register

Note:
1. Each register is placed in a reset state after the clock is provided via the G0BCR0 register.

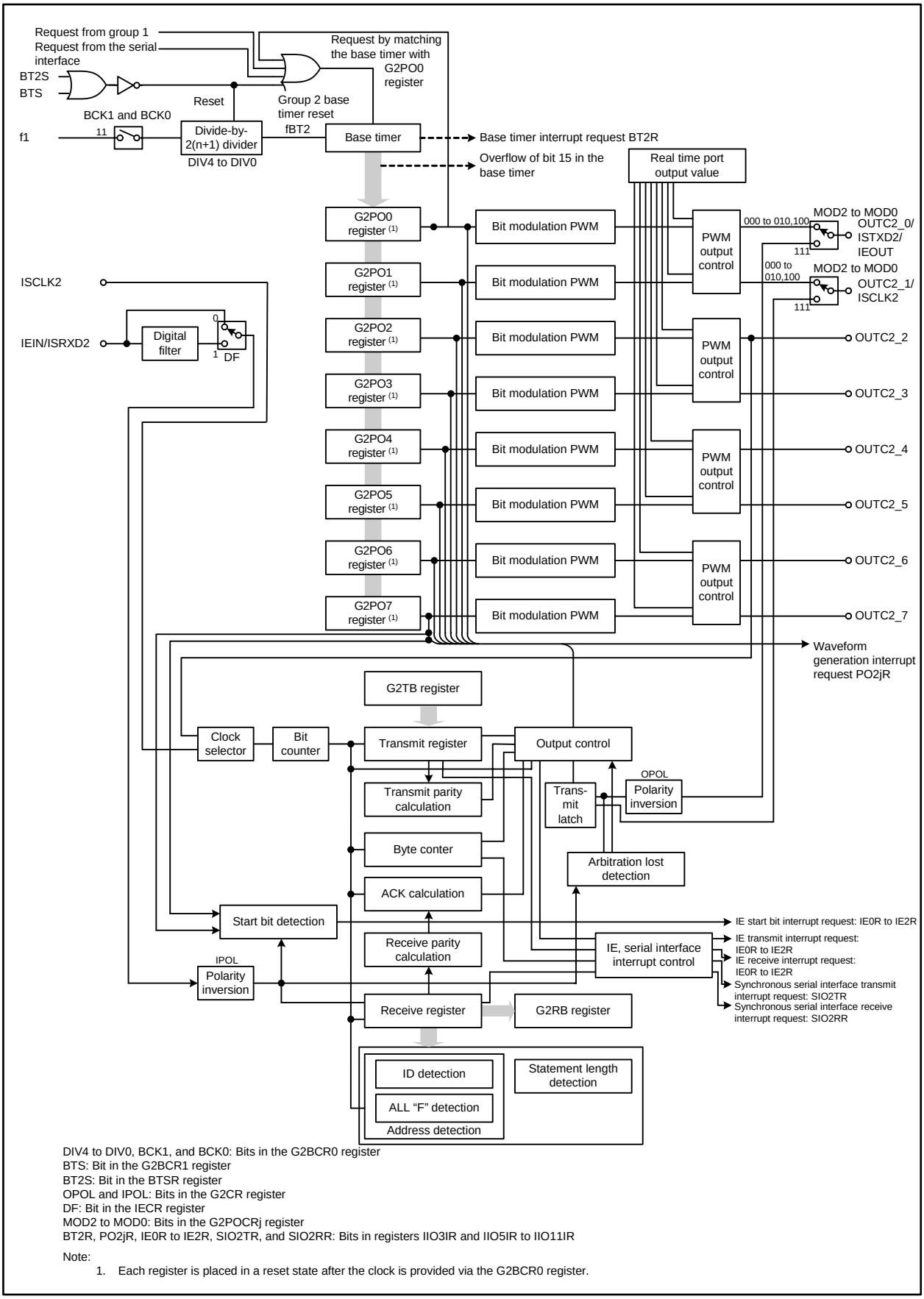
Figure 22.1 Intelligent I/O Group 0 Block Diagram (j = 0 to 7)



DIV4 to DIV0, BCK1, and BCK0: Bits in the G1BCR0 register
BTS: Bit in the G1BCR1 register
BT1S: Bit in the BTSR register
CTS1, CTS0, DF1, DF0, GT, and PR: Bits in the G1TMCRj register
BT1R: Bit in the IIO4IR register

Note:
1. Each register is placed in a reset state after the clock is provided via the G1BCR0 register.

Figure 22.2 Intelligent I/O Group 1 Block Diagram (j = 0 to 7)



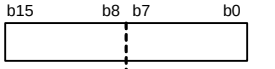
DIV4 to DIV0, BCK1, and BCK0: Bits in the G2BCR0 register
BTS: Bit in the G2BCR1 register
BT2S: Bit in the BTSR register
OPOL and IPOL: Bits in the G2CR register
DF: Bit in the IECR register
MOD2 to MOD0: Bits in the G2POCRj register
BT2R, PO2jR, IE0R to IE2R, SIO2TR, and SIO2RR: Bits in registers IIO3IR and IIO5IR to IIO11IR

Note:
1. Each register is placed in a reset state after the clock is provided via the G2BCR0 register.

Figure 22.3 Intelligent I/O Group 2 Block Diagram (j = 0 to 7)

Figure 22.4 to Figure 22.17 show registers associated with the intelligent I/O base timer, the time measurement, and the waveform generation (For registers associated with the serial interface, refer to Figure 22.33 to Figure 22.40).

Group i Base Timer Register (i = 0 to 2) (1)



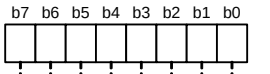
Symbol	Address	Reset Value
G0BT, G1BT	01A1h-01A0h, 0121h-0120h	Undefined
G2BT	0161h-0160h	Undefined

Function	Setting Range	RW
- While the base timer is running, this register indicates the value of base timer; when a value is written, the counter immediately starts counting from this value. The register is set to 0000h when the base timer is reset. - While the base timer is being reset, this register is set to 0000h; the register is read as undefined value; no value can be set. (2)	0000h to FFFFh	RW

Notes:
1. The GiBT register reflects the base timer value after a delay of a half fBTi cycle.
2. The base timer stops only when bits BCK1 and BCK0 in the GiBCR0 register are set to 00b (clock stopped). However, if the BTiS bit in the BTSR register and the BTS bit in the GiBCR1 register are set to 0, the base timer is held in a reset state, holding the value 0000h. That is, the base timer is in a “no counting” state. If either of these bits is set to 1, this state is cleared and the base timer starts counting.

Figure 22.4 Registers G0BT to G2BT

Group i Base Timer Control Register 0 (i = 0 to 2)



Symbol	Address	Reset Value
G0BCR0 to G2BCR0	01A2h, 0122h, 0162h	0000 0000b

Bit Symbol	Bit Name	Function	RW
BCK0	Count Source Select Bit	b1 b2 0 0 : Clock stopped 0 1 : Do not use this combination 1 0 : Two-phase pulse signal input (1) 1 1 : f1	RW
BCK1			RW
DIV0	Count Source Divide Ratio Select Bit	Divide the count source by 2(n+1). The count source is not divided when n = 31. (n = 0 to 31) b6 b5 b4 b3 b2 0 0 0 0 0 : divide-by-2 (n = 0) 0 0 0 0 1 : divide-by-4 (n = 1) 0 0 0 1 0 : divide-by-6 (n = 2) : 1 1 1 1 0 : divide-by-62 (n = 30) 1 1 1 1 1 : no division (n = 31)	RW
DIV1			RW
DIV2			RW
DIV3			RW
DIV4			RW
IT	Base Timer Interrupt Source Select Bit	0: Overflow of bit 15 or bit 9 1: Overflow of bit 14	RW

Note:
1. This bit setting is enabled only when bits UD1 and UD0 in the GjBCR1 register (j = 0, 1) are set to 10b (two-phase pulse signal processing mode). Bits BCK1 and BCK0 should not be set to 10b in other modes or in group 2.

Figure 22.5 Registers G0BCR0 to G2BCR0

Group i Base Timer Control Register 1 (i = 0, 1)

b7	b6	b5	b4	b3	b2	b1	b0
0				0			

Symbol: G0BCR1, G1BCR1 — Address: 01A3h, 0123h — Reset Value: 0000 0000b

Bit Symbol	Bit Name	Function	RW
RST0	Base Timer Reset Source Select Bit 0	0: No reset 1: Synchronization with another base timer reset (1)	RW
RST1	Base Timer Reset Source Select Bit 1	0: No reset 1: Match with the GiPO0 register (2)	RW
RST2	Base Timer Reset Source Select Bit 2	0: No reset 1: A low signal input into the $\overline{INT0}$/$\overline{INT1}$ pin (3)	RW
— (b3)	Reserved	Should be written with 0	RW
BTS	Base Timer Start Bit (4, 5)	0: Reset the base timer 1: Start counting	RW
UD0	Increment/Decrement Counting Control Bit	b6 b5 0 0 : Increment counting mode 0 1 : Increment/decrement counting mode 1 0 : Two-phase pulse signal processing mode (6) 1 1 : Do no use this combination	RW
UD1			RW
— (b7)	Reserved	Should be written with 0	RW

Notes:
1. The group 0 base timer is reset by synchronizing with the reset of group 1 base timer, and vice versa.
2. The base timer is reset after two fBTi clock cycles if the base timer value has matched the GiPO0 register setting (Refer to Figure 23.14 for the details on the GiPO0 register). When the RST1 bit is set to 1, the value of GiPOj register (j = 1 to 7) to be used for the waveform generation should be smaller than that of the GiPO0 register.
3. The base timer is reset by an input of low signal to the external interrupt input pin selected for the UDiZ signal by the IFS2 register.
4. To start base timers groups 0 and 1 individually, the BTS bit should be set to 1 after setting the BTkS bit (k = 0, 1) in the BTSR register to 0 (base timer is reset).
5. To start the base timers of two or all groups simultaneously, the BTSR register should be used. The BTS bit should be set to 0.
6. In two-phase pulse signal processing mode, the base timer is not reset, even though the RST1 bit is set to 1, if the timer counter decrements the count after two clock cycles when the base timer value matches the GiPO0 register.

Figure 22.6 Registers G0BCR1 and G1BCR1

Group 2 Base Timer Control Register 1

b7	b6	b5	b4	b3	b2	b1	b0
	0	0		0			

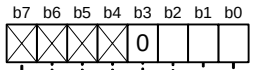
Symbol: G2BCR1 | Address: 0163h | Reset Value: 0000 0000b

Bit Symbol	Bit Name	Function	RW
RST0	Base Timer Reset Source Select Bit 0	0: No reset 1: Synchronization with group1 base timer reset	RW
RST1	Base Timer Reset Source Select Bit 1	0: No reset 1: Match with the G2PO0 register (1)	RW
RST2	Base Timer Reset Source Select Bit 2	0: No reset 1: A reset request from the serial interface	RW
— (b3)	Reserved	Should be written with 0	RW
BTS	Base Timer Start Bit (2, 3)	0: Reset the base timer 1: Start counting	RW
— (b6-b5)	Reserved	Should be written with 0	RW
PRP	Parallel Real-time Port Select Bit (4)	0: RTP output mode 1: Parallel RTP output mode	RW

Notes:
1. The base timer is reset after two fBT2 clock cycles if the base timer value matches the G2PO0 register setting. When the RST1 bit is set to 1, the value of G2POj register (j = 1 to 7) used for the waveform generation or the serial interface should be smaller than that of the G2PO0 register.
2. To start the group 2 base timer, the BTS bit should be set to 1 after setting the BT2S bit in the BTSR register to 0 (base timer is reset).
3. To start the base timers of two or all groups simultaneously, the BTSR register should be used. The BTS bit should be set to 0.
4. This bit setting is enabled when the RTP bit in the G2POCRi register is set to 1 (real-time port used).

Figure 22.7 G2BCR1 Register

Base Timer Start Register (1, 2)



Symbol	Address	Reset Value
BTSR	0164h	XXXX 0000b

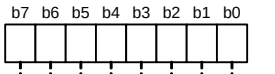
Bit Symbol	Bit Name	Function	RW
BT0S	Group 0 Base Timer Start Bit	0: Reset the base timer 1: Start counting	RW
BT1S	Group 1 Base Timer Start Bit	0: Reset the base timer 1: Start counting	RW
BT2S	Group 2 Base Timer Start Bit	0: Reset the base timer 1: Start counting	RW
— (b3)	Reserved	Should be written with 0	RW
— (b7-b4)	No register bits; should be written with 0 and read as undefined value.		—

Notes:

1. The initial settings of bits and registers for the intelligent I/O are required as follows:
   (1) Set the G2BCR0 register to provide the clock to the group 2 base timer
   (2) Set all bits BT0S to BT2S to 0 (base timer is reset)
   (3) Set other registers associated with the intelligent I/O
   The BTiS bit (i = 0 to 2) allows the base timers of two or all groups to start counting simultaneously. To start counting individually, the BTiS bit should be set to 0 and the BTS bit in the GiBCR1 register should be used.
2. The following procedure is required to start the base timers of more than two groups simultaneously:
   -Bits BCK1 to BCK0 and DIV4 to DIV0 in the GiBCR0 registers (i = two among 0, 1, and 2 or all) to be used should be set identically
   -After bits BCK1 to BCK0 or DIV4 to DIV0 are changed, start the base timers twice as following procedure:
   (1) Set the BTiS bit to 1 (base timer starts counting)
   (2) Set the BTiS bit to 0 (base timer is reset) after one fBTi clock cycle or more
   (3) Set the BTiS bit to 1 again after another one fBTi clock cycle or more

Figure 22.8 BTSR Register

Group i Time Measurement Control Register j (i = 0, 1; j = 0 to 7)



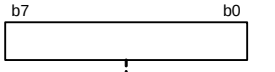
Symbol	Address	Reset Value
G0TMCR0 to G0TMCR3	0198h, 0199h, 019Ah, 019Bh	0000 0000b
G0TMCR4 to G0TMCR7	019Ch, 019Dh, 019Eh, 019Fh	0000 0000b
G1TMCR0 to G1TMCR3	0118h, 0119h, 011Ah, 011Bh	0000 0000b
G1TMCR4 to G1TMCR7	011Ch, 011Dh, 011Eh, 011Fh	0000 0000b

Bit Symbol	Bit Name	Function	RW
CTS0	Time Measurement Trigger Select Bit	b1 b0 0 0 : No time measurement 0 1 : Rising edge 1 0 : Falling edge 1 1 : Both edges	RW
CTS1			RW
DF0	Digital Filter Select Bit	b3 b2 0 0 : No digital filter used 0 1 : Do not use this combination 1 0 : fBTi 1 1 : f1	RW
DF1			RW
GT	Gating Select Bit (1)	0: Gating disabled 1: Gating enabled	RW
GOC	Gating Clear Select Bit (1, 2, 3)	0: Gating not cleared 1: Gating cleared when the base timer matches the GiPOk register (k = j - 2)	RW
GSC	Gating Clear Bit (1, 2)	Gating is cleared by setting this bit to 1	RW
PR	Prescaler Select Bit (1)	0: No prescaler used 1: Use the prescaler	RW

Notes:
1. These functions are available in registers GiTMCR6 and GiTMCR7. Bits 4 to 7 in registers GiTMCR0 to GiTMCR5 should be set to 0.
2. These bit settings are enabled when the GT bit is set to 1.
3. The GOC bit becomes 0 after gating is cleared.

Figure 22.9 Registers G0TMCR0 to G0TMCR7 and G1TMCR0 to G1TMCR7

Group i Time Measurement Prescaler Register j (i = 0, 1; j = 6, 7)



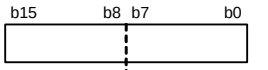
Symbol	Address	Reset Value
G0TPR6, G0TPR7	01A4h, 01A5h	00h
G1TPR6, G1TPR7	0124h, 0125h	00h

Function	Setting Range	RW
Time measurement is executed whenever a trigger input is counted by n+1 (n = setting value) (1)	00h to FFh	RW

Note:
1. The first prescaler, after the PR bit in the GiTMCRj register is changed from 0 (no prescaler is used) to 1 (the prescaler is used), may be divided by n rather than n+1. The subsequent prescaler is divided by n+1.

Figure 22.10 Registers G0TPR6, G0TPR7, G1TPR6 and G1TPR7

Group i Time Measurement Register j (i = 0, 1; j = 0 to 7)

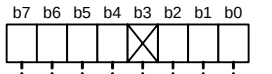


Symbol	Address	Reset Value
G0TM0, G0TM1	0181h-0180h, 0183h-0182h	Undefined
G0TM2, G0TM3	0185h-0184h, 0187h-0186h	Undefined
G0TM4, G0TM5	0189h-0188h, 018Bh-018Ah	Undefined
G0TM6, G0TM7	018Dh-018Ch, 018Fh-018Eh	Undefined
G1TM0, G1TM1	0101h-0100h, 0103h-0102h	Undefined
G1TM2, G1TM3	0105h-0104h, 0107h-0106h	Undefined
G1TM4, G1TM5	0109h-0108h, 010Bh-010Ah	Undefined
G1TM6, G1TM7	010Dh-010Ch, 010Fh-010Eh	Undefined

Function	Setting Range	RW
The base timer value is stored every measurement timing	—	RO

Figure 22.11 Registers G0TM0 to G0TM7 and G1TM0 to G1TM7

Group i Waveform Generation Control Register j (i = 0, 1; j = 0 to 7)



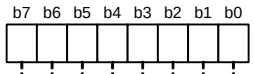
Symbol	Address	Reset Value
G0POCR0	0190h	0000 X000b
G0POCR1 to G0POCR3	0191h, 0192h, 0193h	0X00 X000b
G0POCR4 to G0POCR7	0194h, 0195h, 0196h, 0197h	0X00 X000b
G1POCR0	0110h	0000 X000b
G1POCR1 to G1POCR3	0111h, 0112h, 0113h	0X00 X000b
G1POCR4 to G1POCR7	0114h, 0115h, 0116h, 0117h	0X00 X000b

Bit Symbol	Bit Name	Function	RW
MOD0	Operating Mode Select Bit	b2 b1 b0 0 0 0 : Single-phase waveform output mode 0 0 1 : SR waveform output mode (1) 0 1 0 : Inverted waveform output mode 0 1 1 : Do not use this combination 1 0 0 : Do not use this combination 1 0 1 : Do not use this combination 1 1 0 : Do not use this combination 1 1 1 : Do not use this combination	RW
MOD1			RW
MOD2			RW
— (b3)	No register bit; should be written with 0 and read as undefined value		—
IVL	Default Output Value Select Bit (2)	0: Output low as default value 1: Output high as default value	RW
RLD	GiPOj Register Value Reload Timing Select Bit	0: Reload the value into the GiPOj register on a write access 1: Reload the value into the GiPOj register when the base timer is reset	RW
BTRE	Base Timer Reset Enable Bit (3)	0: Reset the base timer when bit 15 overflows 1: Reset the base timer when bit 9 overflows (4)	RW
INV	Output Level Inversion Select Bit (5)	0: Do not invert the output level 1: Invert the output level	RW

Notes:
1. This bit setting is enabled only for even channels. In SR waveform output mode, the corresponding odd channel (the next channel after an even channel) setting is ignored. Waveforms are only output from even channels.
2. The setting value is output by a write to the IVL bit if the FSCj bit in the GiFS register is set to 0 (the waveform generation selected) and the IFEj bit in the GiFE register is set to 1 (the function for channel j enabled).
3. This bit is available only in the GiPOCR0 register. Bit 6 in registers GiPOCR1 to GiPOCR7 should be set to 0.
4. To set the BTRE bit to 1, bits BCK1 and BCK0 in the GiBCR0 register should be set to 11b (f1) and bits UD1 and UD0 in the GiBCR1 register should be set to 00b (increment counting mode).
5. The output level inversion is the final step in waveform generation process. When the INV bit is set to 1 (output level is inverted), high is output by setting the IVL bit to 0 (output low as default value), and vice versa.

Figure 22.12 Registers G0POCR0 to G0POCR7 and G1POCR0 to G1POCR7

Group 2 Waveform Generation Control Register j (j = 0 to 7)



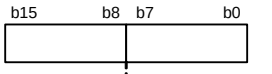
Symbol	Address	Reset Value
G2POCR0 to G2POCR3	0150h, 0151h, 0152h, 0153h	0000 0000b
G2POCR4 to G2POCR7	0154h, 0155h, 0156h, 0157h	0000 0000b

Bit Symbol	Bit Name	Function	RW
MOD0	Operating Mode Select Bit (1)	b2 b1 b0 0 0 0 : Single waveform output mode 0 0 1 : SR waveform output mode (2) 0 1 0 : Inverted waveform output mode 0 1 1 : Do not use this combination 1 0 0 : Bit modulation PWM output mode 1 0 1 : Do not use this combination 1 1 0 : Do not use this combination 1 1 1 : Use an output for the serial interface (3)	RW
MOD1			RW
MOD2			RW
PRT	Parallel Real-time Port Output Trigger Select Bit (4)	0: Not triggered by matching the base timer with registers G2PO0 to G2PO7 1: Triggered by matching the base timer with registers G2PO0 to G2PO7	RW
IVL	Default Output Value Select Bit	0: Output low as default value 1: Output high as default value	RW
RLD	G2POj Register Value Reload Timing Select Bit	0: Reload the value into the G2POj register on a write access 1: Reload the value into the G2POj register when the base timer is reset	RW
RTP	Real-time Port Select Bit	0: No real-time port function used 1: Use RTP output mode or parallel RTP output mode	RW
INV	Output Level Inversion Select Bit (5)	0: Do not invert the output level 1: Invert the output level	RW

Notes:
1. When the RTP bit is set to 1, the settings of bits MOD2 to MOD0 are disabled.
2. This bit setting is enabled only for even channels. In SR waveform output mode, the corresponding odd channel (the next channel after an even channel) setting is ignored. Waveforms are not output from odd channels but even channels.
3. This bit setting is enabled only for channels 0 and 1 of the group 2. To use the ISTXD2 or IEOUT pin as an output, bits MOD2 to MOD0 in the G2POCR0 register should be set to 111b. To use the ISCLK2 pin, the same bits in the G2POCR1 register should be set to 111b. This bit setting should not be done in other channels than 0 and 1.
4. This bit setting is enabled when the RTP bit is set to 1 (real-time port used) and the PRP bit in the G2BCR1 register is set to 1 (parallel RTP output mode).
5. The output level inversion is the final step in waveform generation process. When the INV bit is set to 1 (output level is inverted), high is output by setting the IVL bit to 0 (output low as default value), and vice versa.

Figure 22.13 Registers G2POCR0 to G2POCR7

Group i Waveform Generation Register j (i = 0 to 2; j = 0 to 7)

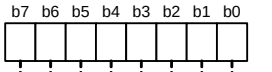


Symbol	Address	Reset Value
G0PO0 to G0PO2	0181h-0180h, 0183h-0182h, 0185h-0184h	Undefined
G0PO3 to G0PO5	0187h-0186h, 0189h-0188h, 018Bh-018Ah	Undefined
G0PO6, G0PO7	018Dh-018Ch, 018Fh-018Eh	Undefined
G1PO0 to G1PO2	0101h-0100h, 0103h-0102h, 0105h-0104h	Undefined
G1PO3 to G1PO5	0107h-0106h, 0109h-0108h, 010Bh-010Ah	Undefined
G1PO6, G1PO7	010Dh-010Ch, 010Fh-010Eh	Undefined
G2PO0 to G2PO2	0141h-0140h, 0143h-0142h, 0145h-0144h	Undefined
G2PO3 to G2PO5	0147h-0146h, 0149h-0148h, 014Bh-014Ah	Undefined
G2PO6, G2PO7	014Dh-014Ch, 014Fh-014Eh	Undefined

Function	Setting Value	RW
-If the RLD bit in the GiPOCRj register is set to 0, the value is reloaded into the GiPOj register immediately after being written to be reflected in output waveform -If the RLD bit is set to 1, the value is reloaded when the base timer is reset. The register indicates the written value until the value is reloaded	0000h to FFFFh	RW

Figure 22.14 Registers G0PO0 to G0PO7, G1PO0 to G1PO7, and G2PO0 to G2PO7

Group i Function Select Register (i = 0, 1)

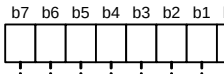


Symbol	Address	Reset Value
G0FS, G1FS	01A7h, 0127h	0000 0000b

Bit Symbol	Bit Name	Function	RW
FSC0	Channel 0 Time Measurement/Waveform Generation Select Bit	0: Select the waveform generation 1: Select the time measurement	RW
FSC1	Channel 1 Time Measurement/Waveform Generation Select Bit		RW
FSC2	Channel 2 Time Measurement/Waveform Generation Select Bit		RW
FSC3	Channel 3 Time Measurement/Waveform Generation Select Bit		RW
FSC4	Channel 4 Time Measurement/Waveform Generation Select Bit		RW
FSC5	Channel 5 Time Measurement/Waveform Generation Select Bit		RW
FSC6	Channel 6 Time Measurement/Waveform Generation Select Bit		RW
FSC7	Channel 7 Time Measurement/Waveform Generation Select Bit		RW

Figure 22.15 Registers G0FS and G1FS

Group i Function Enable Register (i = 0 to 2)

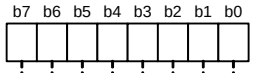


Symbol	Address	Reset Value
G0FE to G2FE	01A6h, 0126h, 0166h	0000 0000b

Bit Symbol	Bit Name	Function	RW
IFE0	Channel 0 Function Enable Bit	0: Disable the function for channel j 1: Enable the function for channel j (j = 0 to 7)	RW
IFE1	Channel 1 Function Enable Bit		RW
IFE2	Channel 2 Function Enable Bit		RW
IFE3	Channel 3 Function Enable Bit		RW
IFE4	Channel 4 Function Enable Bit		RW
IFE5	Channel 5 Function Enable Bit		RW
IFE6	Channel 6 Function Enable Bit		RW
IFE7	Channel 7 Function Enable Bit		RW

Figure 22.16 Registers G0FE to G2FE

Group 2 RTP Output Buffer Register



Symbol	Address	Reset Value
G2RTP	0167h	0000 0000b

Bit symbol	Bit Name	Function	RW
RTP0	Channel 0 RTP Output Buffer	0: Output a low level 1: Output a high level	RW
RTP1	Channel 1 RTP Output Buffer		RW
RTP2	Channel 2 RTP Output Buffer		RW
RTP3	Channel 3 RTP Output Buffer		RW
RTP4	Channel 4 RTP Output Buffer		RW
RTP5	Channel 5 RTP Output Buffer		RW
RTP6	Channel 6 RTP Output Buffer		RW
RTP7	Channel 7 RTP Output Buffer		RW

Figure 22.17 G2RTP Register

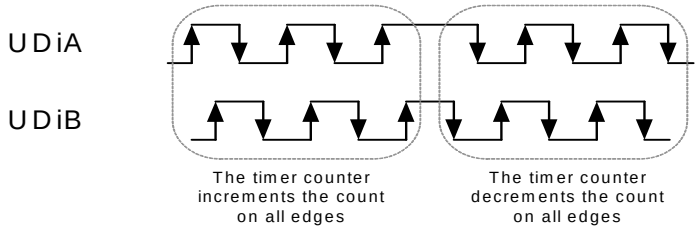
22.1 Base Timer (for Groups 0 to 2)

The base timer is a free-running counter that counts an internally generated count source. Table 22.2 lists specifications of the base timer. Figure 22.4 to Figure 22.17 show registers associated with the base timer. Figure 22.18 shows a block diagram of the base timer. Figure 22.19, Figure 22.20, and Figure 22.21 show respectively an operation example of the base timer (for groups 0 and 1) in increment counting mode, in increment/decrement counting mode, and in two-phase pulse signal processing mode.

Table 22.2 Base Timer Specifications (i = 0 to 2)

Item	Specification
Count source (fBTi)	f1 divided by 2(n+1) (for groups 0 to 2), two-phase pulse input divided by 2(n+1) (for groups 0 and 1) n: setting value using bits DIV4 to DIV0 in the GiBCR0 register n = 0 to 31; however no division when n = 31
Count operations	• Increment counting • Increment/decrement counting • Two-phase pulse signal processing
Count start conditions	• To start each base timer individually, the BTS bit in the GiBCR1 register is set to 1 (count starts) • To start base timers of two or all groups simultaneously, the BTiS bit in the BTSR register is set to 1 (count starts)
Count stop condition	The BTiS bit in the BTSR register is set to 0 (base timer is reset) and the BTS bit in the GiBCR1 register is set to 0 (base timer is reset)
Reset conditions	• The base timer value matches the GiPO0 register setting • An input of low signal into the external interrupt pin as follows: for group 0: selected using bits IFS23 and IFS22 in the IFS2 register for group 1: selected using bits IFS27 and IFS26 in the IFS2 register • The overflow of bit 15 or 9 in the base timer • The base timer reset request from the communication functions (group 2)
Reset value	0000h
Interrupt request	When the BTiR bit in the interrupt request register is set to 1 (interrupts requested) by the overflow of bit 9, 14, or 15 in the base timer (refer to Figure 10.12)
Read from base timer	• The GiBT register indicates a counter value while the base timer is running • The GiBT register is undefined while the base timer is being reset
Write to base timer	When a value is written while the base timer is running, the timer counter immediately starts counting from this value. No value can be written while the base timer is being reset

Table 22.2 Base Timer Specifications (i = 0 to 2)

Item	Specification
Selectable functions	• Increment/decrement counting mode (for groups 0 and 1) The base timer starts counting when the BTS or BTiS bit is set to 1. On reaching FFFFh, it starts decrement counting. When the RST1 bit in the GiBCR1 register is set to 1 (reset by match with the GiPO0 register), the timer counter starts decrement counting as soon as the base timer value has matched the GiPO0 register setting. When the timer counter has reached 0000h, it starts increment counting again (Refer to Figure 22.20). • Two-phase pulse signal processing mode (for groups 0 and 1) Two-phase pulse signals at pins UDiA and UDiB are counted (Refer to Figure 22.21). 

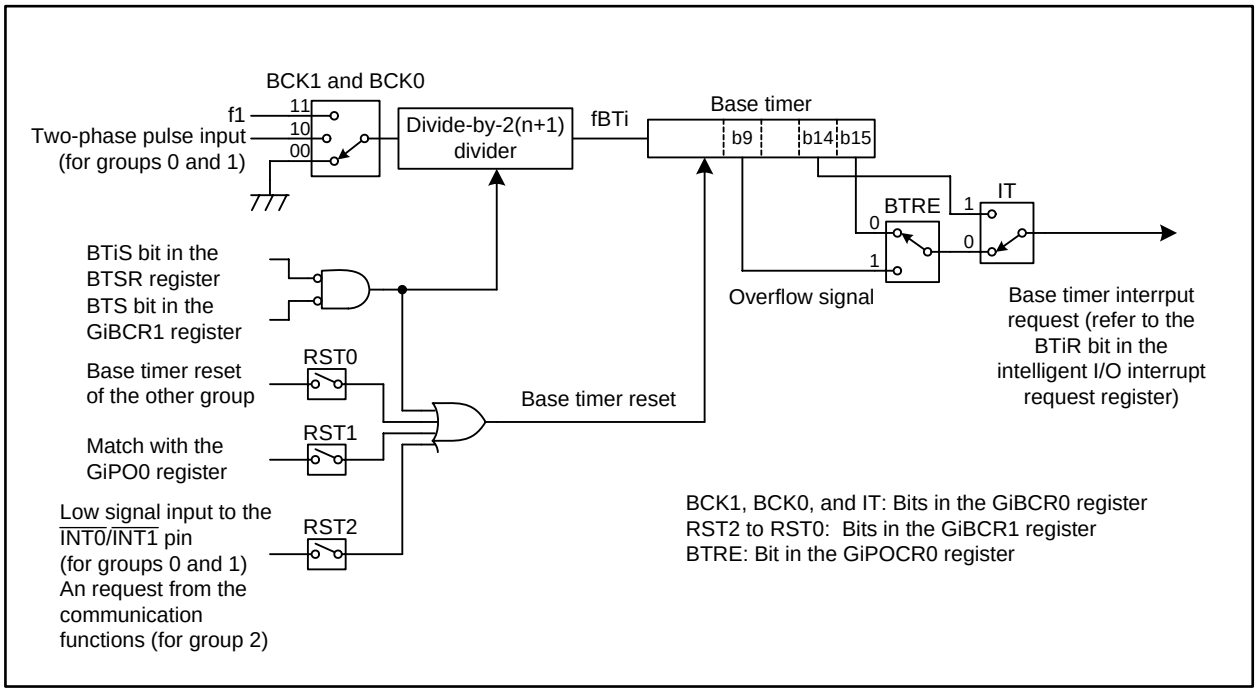

BCK1 and BCK0
f1
Two-phase pulse input (for groups 0 and 1)
11
10
00
Divide-by-2(n+1) divider
fBTi
Base timer
b9
b14
b15
BTRE
IT
BTiS bit in the BTSR register
BTS bit in the GiBCR1 register
Overflow signal
Base timer interrput request (refer to the BTiR bit in the intelligent I/O interrupt request register)
Base timer reset of the other group
RST0
Base timer reset
Match with the GiPO0 register
RST1
Low signal input to the INT0/INT1 pin (for groups 0 and 1)
An request from the communication functions (for group 2)
RST2
BCK1, BCK0, and IT: Bits in the GiBCR0 register
RST2 to RST0: Bits in the GiBCR1 register
BTRE: Bit in the GiPOCR0 register

Figure 22.18 Base Timer Block Diagram (i = 0 to 2)

Table 22.3 Base Timer Associated Register Settings (Common Settings for Time Measurement, Waveform Generation, and Serial Interface) (i = 0 to 2)

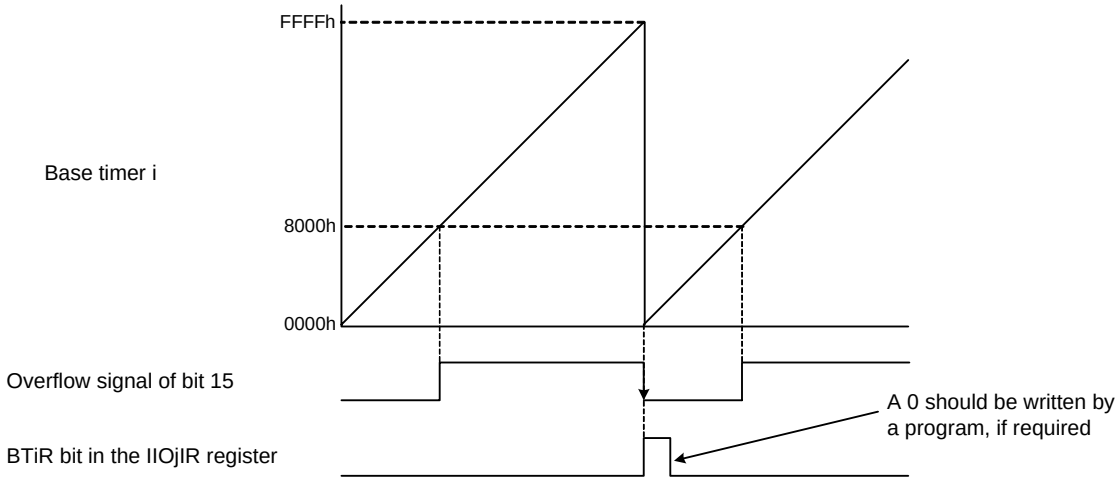
Register	Bits	Function
G2BCR0	—	Provide an operating clock to the BTSR register. Set to 0111 1111b
BTSR	—	Set to 0000 0000b
GiBCR0	BCK1 and BCK0	Select a count source
	DIV4 to DIV0	Select a divide ratio of count source
	IT	Select a base timer interrupt source
GiBCR1	RST2 to RST0	Select a timing for base timer reset
	BTS	Use this bit when each base timer individually starts counting
	UD1 and UD0	Select a count operation (in groups 0 and 1)
GiPOCR0	BTRE	Select a source for base timer reset
GiBT	—	Read or write the base timer value

The following register settings are required to set the RST1 bit to 1 (base timer is reset by match with the GiPO0 register).

GiPOCR0	MOD2 to MOD0	Set to 000b (single-phase waveform output mode)
GiPO0	—	Set the reset cycle
GiFS	FSC0	Set the bit to 0 (waveform generation)
GiFE	IFE0	Set the bit to 1 (channel operation starts)

Bit configurations and functions vary with the groups.

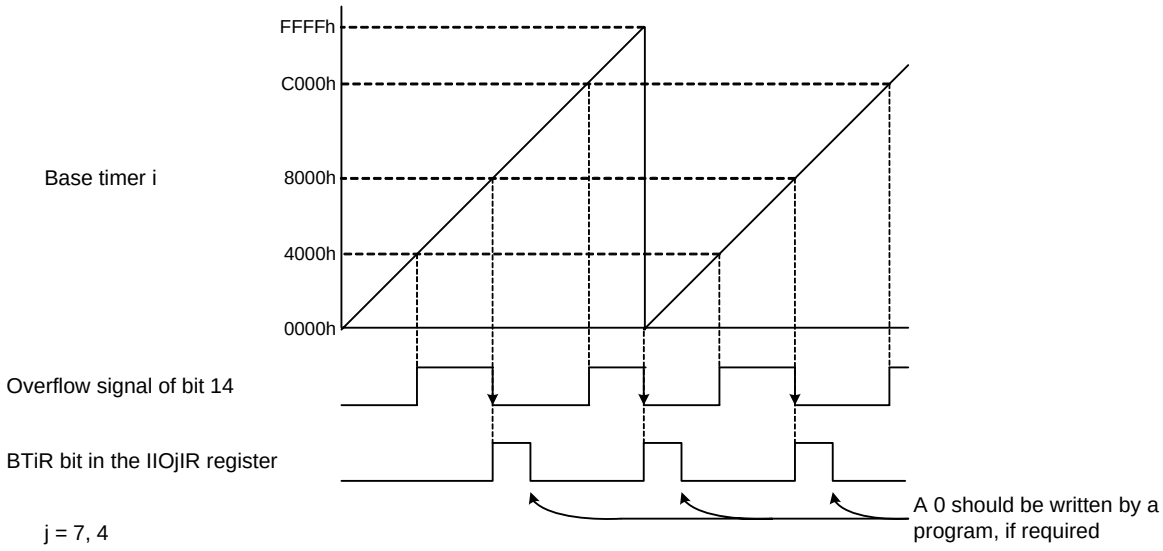
(1) When the IT bit in the GiBCR0 register is set to 0
(an interrupt is requested by the overflow of bit 15 in the base timer)



j = 7, 4

The figure above applies under the following conditions:
- The RST1 bit in the GiBCR1 register is set to 0 (the match with the GiPO0 register is not the reset source for the base timer)
- Bits UD1 and UD0 in the GiBCR1 register are set to 00b (increment counting mode)

(2) When the IT bit in the GiBCR0 register is set to 1
(an interrupt is requested by the overflow of bit 14 in the base timer)

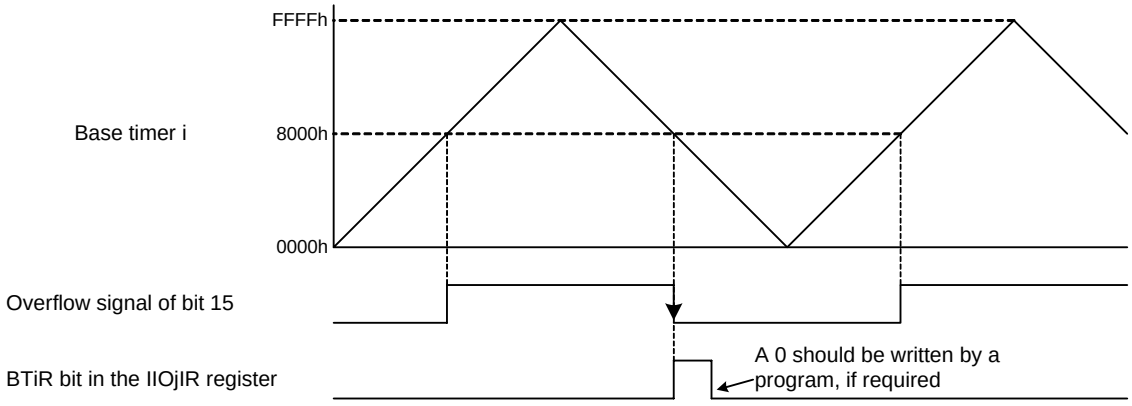


j = 7, 4

The figure above applies under the following conditions:
- The RST1 bit in the GiBCR1 register is set to 0 (the match with the GiPO0 register is not the reset source for the base timer)
- Bits UD1 and UD0 in the GiBCR1 register are set to 00b (increment counting mode)

Figure 22.19 Base Timer Increment Counting Mode (i = 0, 1) (for Groups 0 and 1)

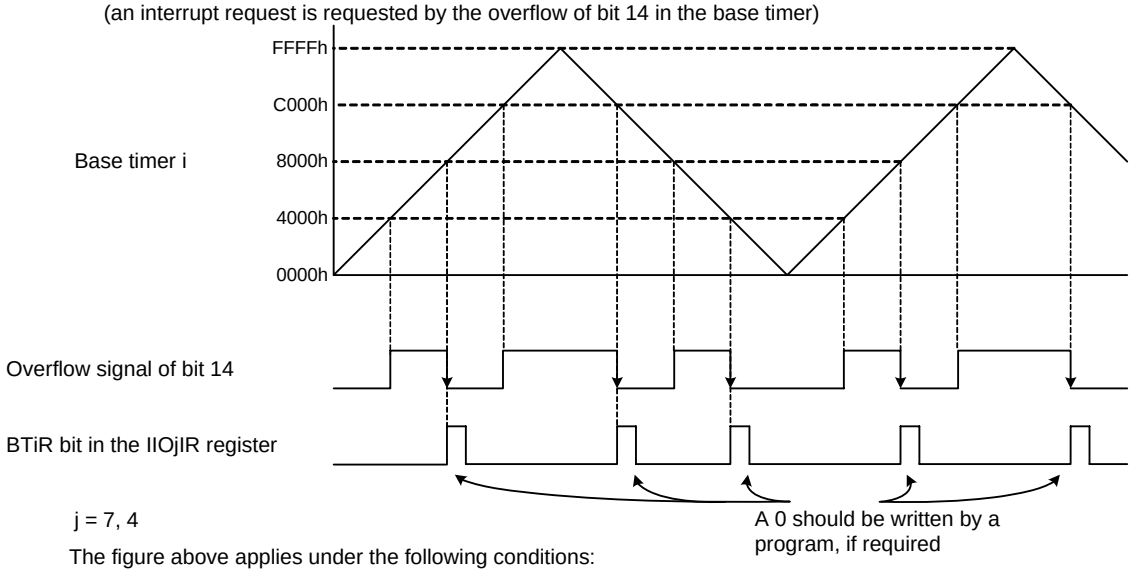
(1) When the IT bit in the GiBCR0 register is set to 0
(an interrupt is requested by the overflow of bit 15 in the base timer)



j = 7, 4

The figure above applies under the following conditions:
- The RST1 bit in the GiBCR1 register is set to 0 (the match with the GiPO0 register is not the reset source for the base timer)
- Bits UD1 and UD0 in the GiBCR1 register are set to 01b (increment/decrement counting mode)

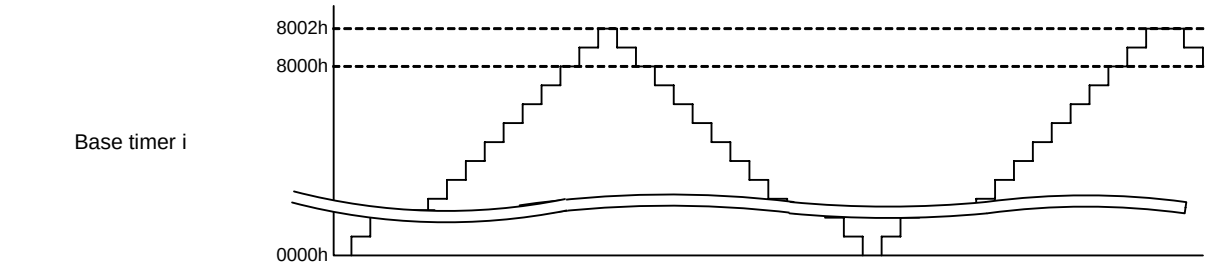
(2) When the IT bit in the GiBCR0 register is set to 1
(an interrupt request is requested by the overflow of bit 14 in the base timer)



j = 7, 4

The figure above applies under the following conditions:
- The RST1 bit in the GiBCR1 register is set to 0 (the match with the GiPO0 register is not the reset source for the base timer)
- Bits UD1 and UD0 in the GiBCR1 register are set to 01b (increment/decrement counting mode)

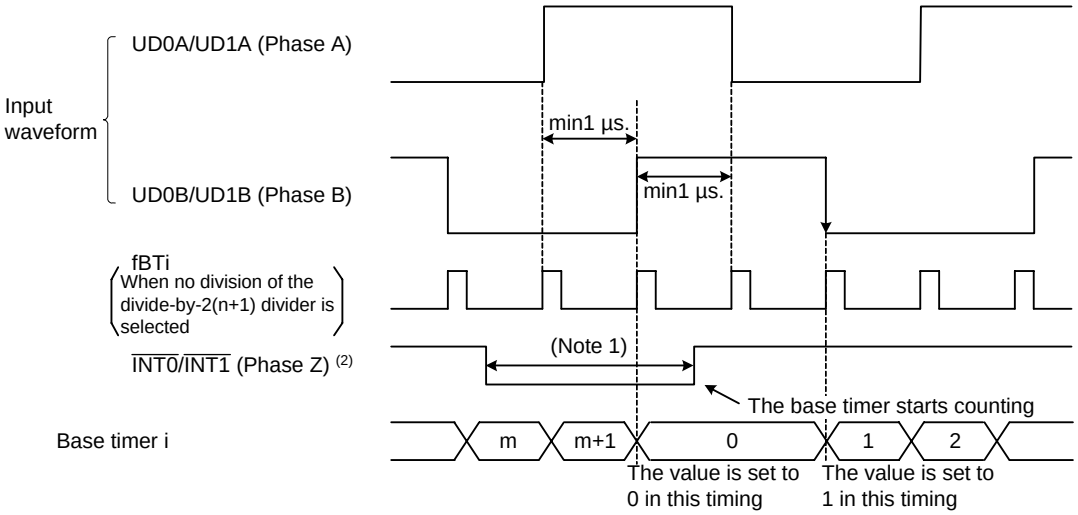
(3) When the RST bit in the GiBCR1 register is set to 1 (the base timer is reset by matching with the GiPO0 register)



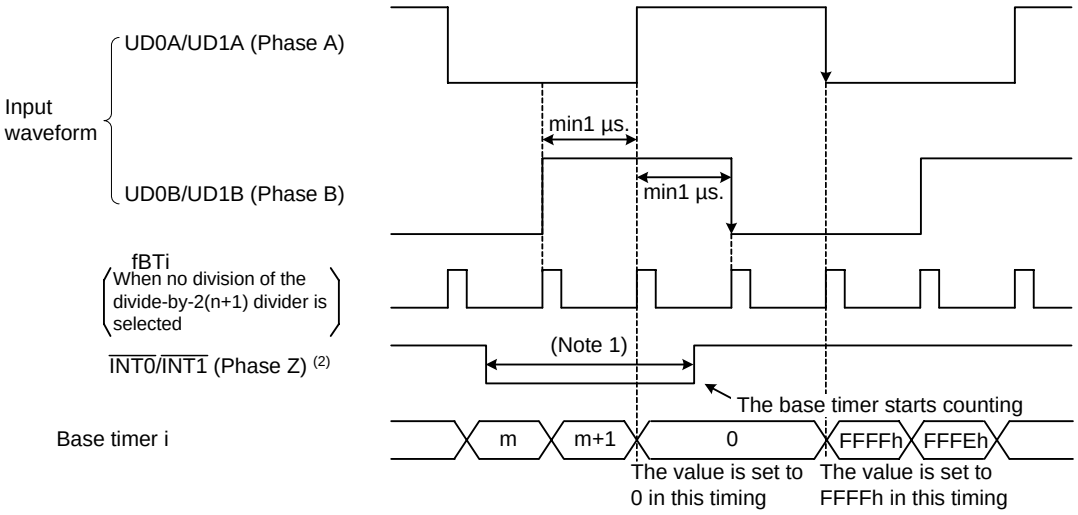
The figure above applies under the following conditions:
- The GiPO0 register value is set to 8000h
- Bits UD1 and UD0 in the GiBCR1 register are set to 01b (increment/decrement counting mode)

Figure 22.20 Base Timer Increment/Decrement Counting (i = 0, 1) (for Groups 0 and 1)

(1) When the base timer is reset while it increments the count



(2) When the base timer is reset while it decrements the count



Notes:
1. 1.5 fBTi clock cycles or more are required.
2. The RST2 bit in the GiBCR1 register should be set to 1 in two-phase pulse signal processing mode.

Figure 22.21 Base Timer Two-phase Pulse Signal Processing Mode (i = 0, 1) (for Groups 0 and 1)

22.2 Time Measurement (for Groups 0 and 1)

Every time an external trigger is input, the base timer value is stored into the GiTMj register (i = 0, 1; j = 0 to 7). Table 22.4 lists specifications of the time measurement and Table 22.5 lists its register settings. Figure 22.22 and Figure 22.23 show operation examples of the time measurement and Figure 22.24 shows operation examples with the prescaler or gate function.

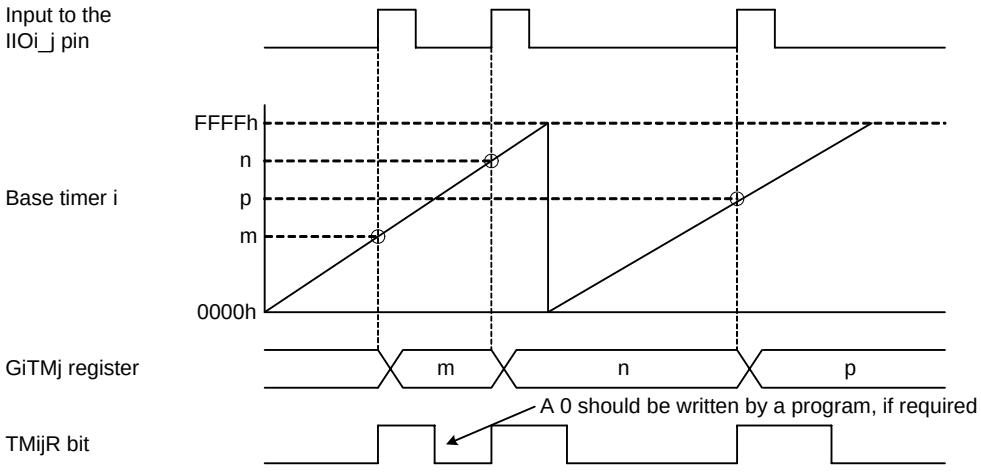
Table 22.4 Time Measurement Specifications (i = 0, 1; j = 0 to 7)

Item	Specification
Time measurement channels	Group 0: Channels 0 to 7 Group 1: Channels 0 to 7
Trigger input polarity	Rising edge, falling edge, or both edges of the IIOi_j pin
Time measurement start condition	The IFEj bit in the GiFE register is set to 1 (function for channel j enabled) while the FSCj bit in the GiFS register is set to 1 (time measurement selected)
Time measurement stop condition	The IFEj bit is set to 0 (function for channel j disabled)
Time measurement timing	• Without the prescaler: every time a trigger is input • With the prescaler (for channels 6 and 7): every (GiTPRk register (k = 6, 7) value + 1) times a trigger is input
Interrupt request	When the TMijR bit in the interrupt request register is set to 1 (interrupts requested) (Refer to Figure 10.12)
IIOi_j input pin function	Trigger input
Selectable functions	• Digital filter The digital filter determines a trigger input level every f1 or fBTi cycle and passes the signals holding the same level during three sequential cycles • Prescaler (for channels 6 and 7) Time measurement is executed every (GiTPRk register value + 1) times a trigger is input • Gating (for channels 6 and 7) This function disables any trigger input to be accepted after the time measurement by the first trigger input. However, the trigger input can be accepted again if any of following conditions are met while the GOC bit in the GiTMCRk register is set to 1 (the gating is cleared when the base timer matches the GiPOp register) (p = 4, 5; p = 4 when k = 6; p = 5 when k = 7): • The base timer value matches the GiPOp register setting • The GSC bit in the GiTMCRk register is set to 1

Table 22.5 Time Measurement (for Groups 0 and 1) Associated Register Settings (i = 0, 1; j = 0 to 7; k = 6, 7)

Register	Bits	Function
GiTMCRj	CTS1 and CTS0	Select a time measurement trigger
	DF1 and DF0	Select a digital filter
	GT, GOC, GSC	Select if the gating is used
	PR	Select if the prescaler is used
GiTPRk	—	Set the prescaler value
GiFS	FSCj	Set the bit to 1 (the time measurement selected)
GiFE	IFEj	Set the bit to 1 (function for channel j enabled)

Bit configurations and functions vary with the channels or groups.
Registers associated with the time measurement should be set after setting the base timer-associated registers.



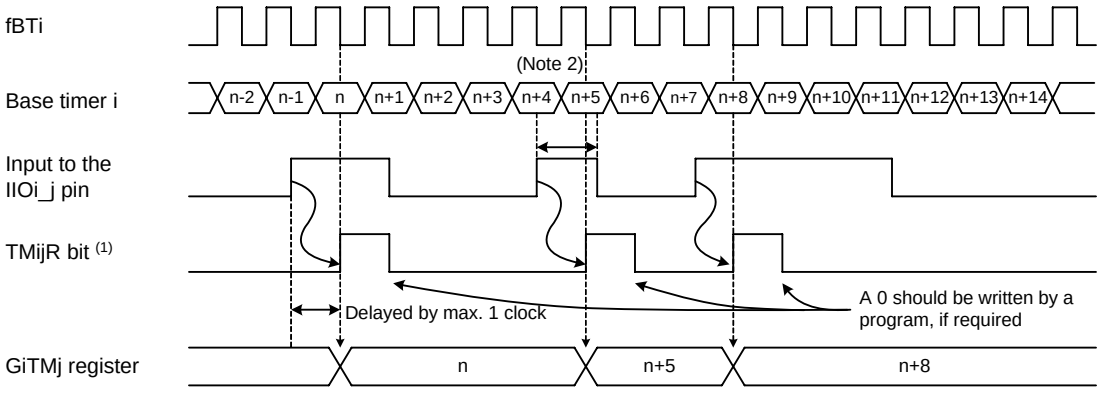
TMijR bit: Bit in registers IIO0IR to IIO11IR

This figure above applies under the following conditions:
- Bits CTS1 and CTS0 in the GiTMCRj register are set to 01b (rising edge as time measurement trigger), the PR bit is set to 0 (no prescaler is used) and the GT bit is set to 0 (no gating is used)
- Bits RST2 to RST0 in the GiBCR1 register are set to 000b (base timer is not reset) and bits UD1 and UD0 are set to 00b (increment counting mode)

The base timer reset by matching the GiPO0 register setting (bits RST2 to RST0 in the GiBCR1 register are set to 010b) is done after the base timer reaches the GiPO0 register setting value + 2.

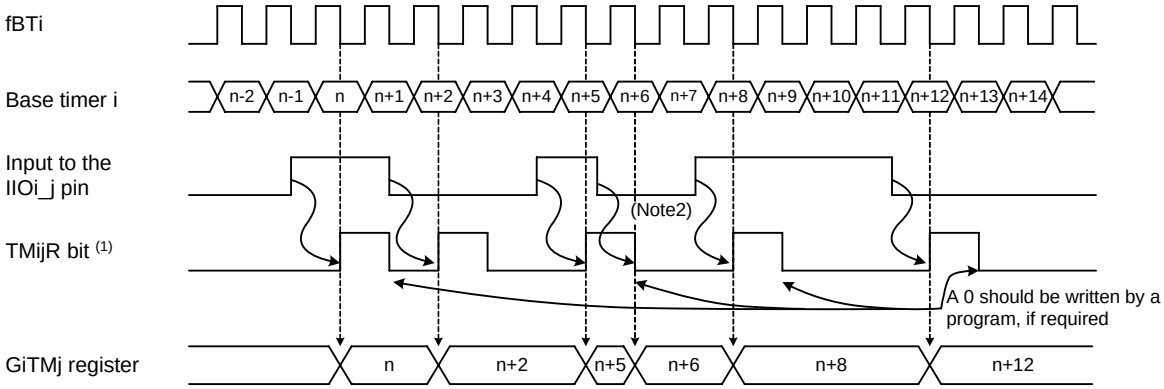
Figure 22.22 Time Measurement Operation (1) (i = 0, 1; j = 0 to 7)

Case 1: When selecting the rising edge as a time measurement trigger
(Bits CTS1 and CTS0 in the GiTMCRj register are set to 01b)



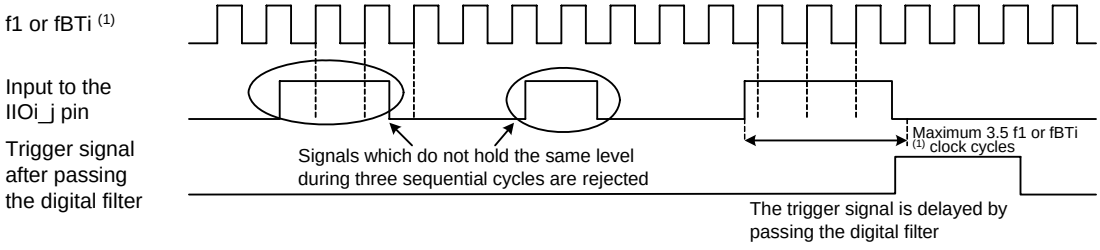
Notes:
1. Bit in registers IIO0IR to IIO11IR
2. Input pulse applied to the IIOi_j pin requires 1.5 fBTi clock cycles or more.

Case 2: When selecting both edges as a time measurement trigger
(Bits CTS1 and CTS0 in the GiTMCRj register are set to 11b)



Notes:
1. Bit in registers IIO0IR to IIO11IR
2. No interrupt is generated if the MCU receives a trigger signal when the TMijR bit is set to 1. However, the value of GiTMj register changes.

Case 3: Trigger signal when using the digital filter
(Bits DF1 and DF0 in the GiTMCRj register are set to 10b or 11b)

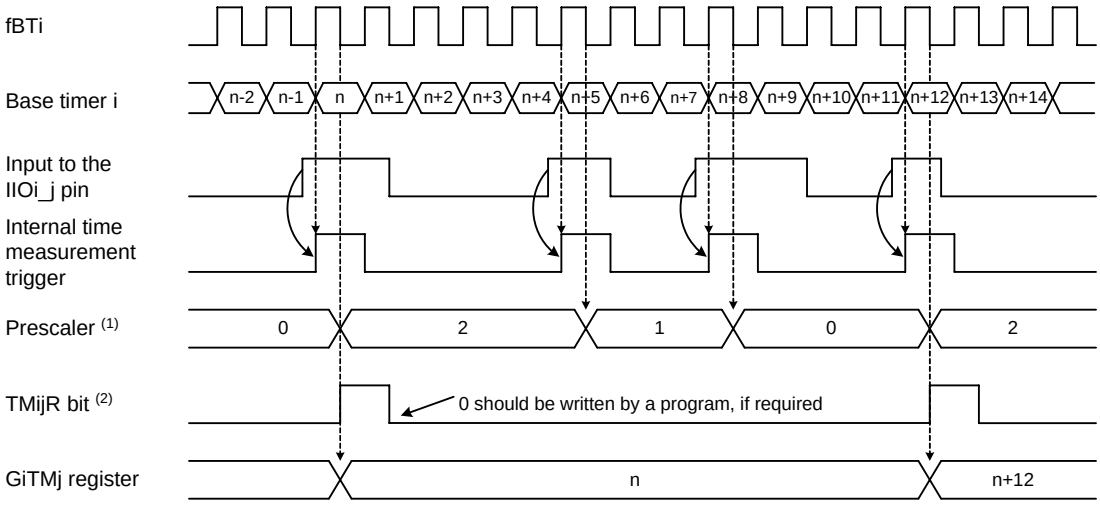

f1 or fBTi (1)
Input to the IIOi_j pin
Trigger signal after passing the digital filter
Signals which do not hold the same level during three sequential cycles are rejected
Maximum 3.5 f1 or fBTi (1) clock cycles
The trigger signal is delayed by passing the digital filter

Note:
1. fBTi when bits DF1 and DF0 are set to 10b, f1 when the bits are set to 11b.

Figure 22.23 Time Measurement Operation (2) (i = 0, 1; j = 0 to 7)

Case 1: Operation with the prescaler

(The GiTPRj register is set to 02h and the PR bit in the GiTMCRj register is set to 1)

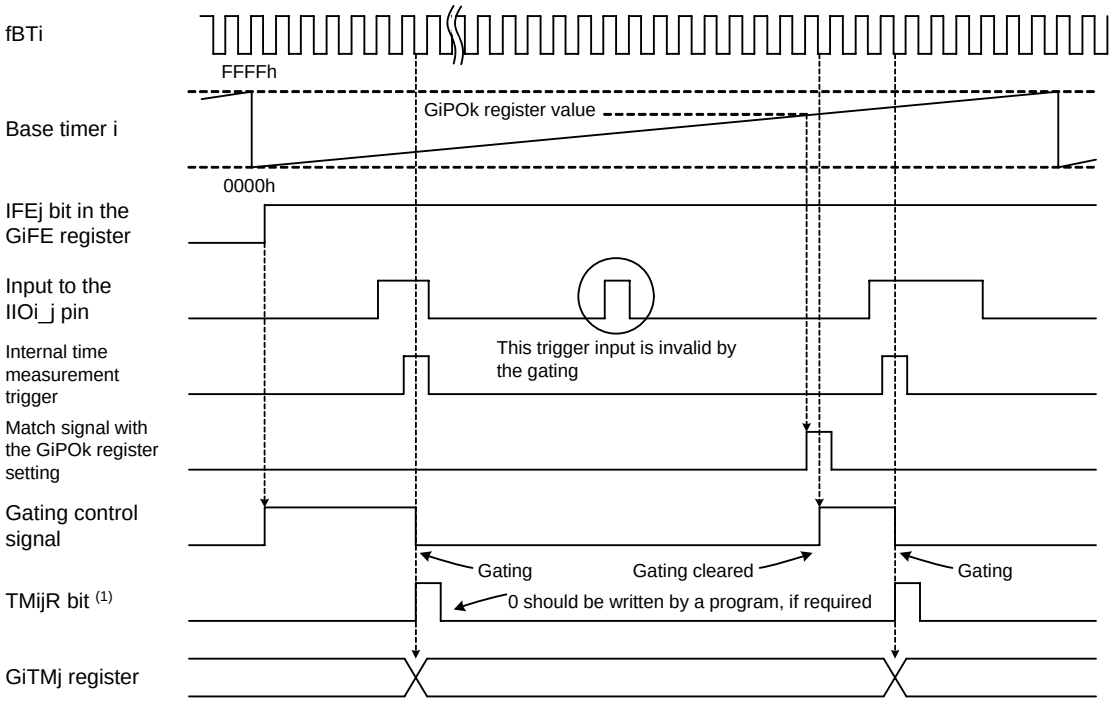


Notes:

1. This example applies to cycles following the first cycle after the PR bit in the GiTMCRj register is set to 1 (the prescaler is used)
2. Bit in registers IIO0IR to IIO11IR

Case 2: Operation with the gating

(The gating is cleared by matching the base timer value with the GiPOk register setting. Bits GT and GOC in the GiTMCRj register are respectively set to 1)



Note:

1. Bit in registers IIO0IR to IIO11IR

Figure 22.24 Prescaler and Gate Operations (i = 0, 1; j = 6, 7; k = 4, 5)

22.3 Waveform Generation (for Groups 0 to 2)

Waveforms are generated when the base timer value matches the GiPOj register setting (i = 0 to 2; j = 0 to 7).

Waveform generation has the following six modes:

- Single-phase waveform output mode (for groups 0 to 2)
- Inverted waveform output mode (for groups 0 to 2)
- Set/reset waveform output (SR waveform output) mode (for groups 0 to 2)
- Bit modulation PWM output mode (for group 2)
- Real-time port output (RTP output) mode (for group 2)
- Parallel real-time port output (parallel RTP output) mode (for group 2)

Table 22.6 lists registers associated with the waveform generation.

Table 22.6 Waveform Generation Associated Register Settings (i = 0 to 2; j = 0 to 7)

Register	Bits	Function
GiPOCRj	MOD2 to MOD0	Select a waveform output mode
	PRT (1)	Set the bit to 1 to use parallel RTP output mode
	IVL	Select a default value
	RLD	Select a timing to reload the value into the GiPOj register
	RTP (1)	Set the bit to 1 to use RTP output mode or parallel RTP output mode. The settings of bits MOD2 to MOD0 are disabled when this bit is set to 1
	INV	Select if output level is inverted
G2BCR1	PRP	Set the bit to 1 to use parallel RTP output mode
GiPOj	—	Set the timing to invert output waveform
GiFS	FSCj	Set the bit to 0 (the waveform generation selected) (for groups 0 and 1 only)
GiFE	IFEj	Set the bit to 1 (the function for channel j enabled)
G2RTP	RTP0 to RTP7	Set the RTP output value in RTP output mode or parallel RTP output mode

Bit configurations and functions vary with channels or groups.

Registers associated with the waveform generation should be set after setting the base timer-associated registers.

Note:

1. This bit is available in the G2POCRj register only. Neither the G0POCRj nor G1POCRj register has it.

22.3.1 Single-phase Waveform Output Mode (for Groups 0 to 2)

The output level at the IIOi_j pin (or OUTC2_j pin for Group 2) becomes high when the base timer value matches the GiPOj register (i = 0 to 2; j = 0 to 7). It switches to low when the base timer reaches 0000h. If the IVL bit in the GiPOCRj register is set to 1 (high level is output as default value), a high level output is provided when a waveform output starts. If the INV bit is set to 1 (output level is inverted), a waveform with inverted level is output. Refer to Figure 22.25 for details on single-phase waveform mode operation.

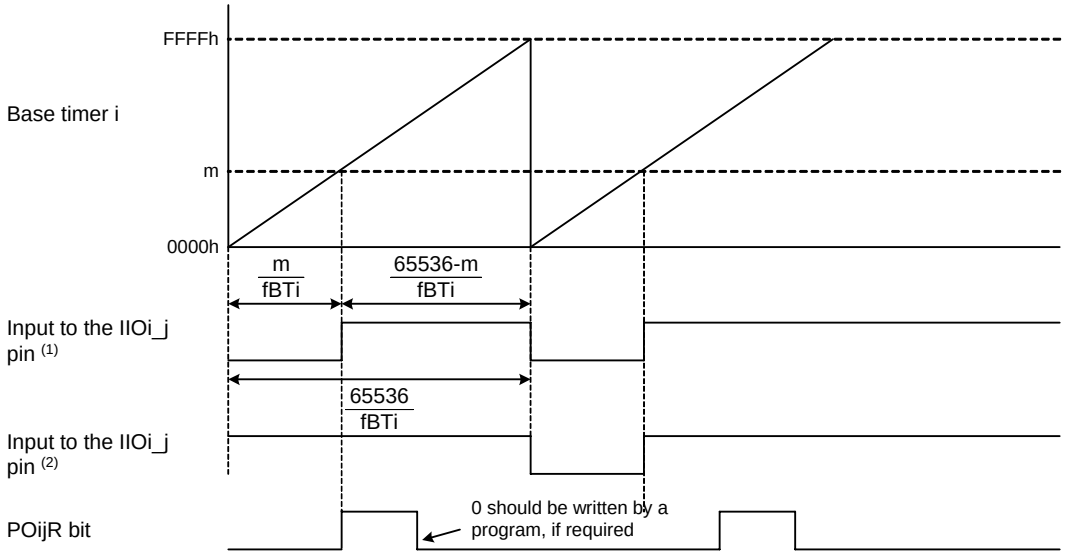
Table 22.7 lists specifications of single-phase waveform output mode.

Table 22.7 Single-phase Waveform Output Mode Specifications (i= 0 to 2)

Item	Specification
Output waveform (1)	• Free-running operation (when bits RST2 to RST0 in the GiBCR1 register are set to 000b) Cycle: $\frac{65536}{fBTi}$ Low level width: $\frac{m}{fBTi}$ High level width: $\frac{65536 - m}{fBTi}$ m: GiPOj register (j = 0 to 7) setting value, 0000h to FFFFh • The base timer is reset by matching the base timer value with the GiPO0 register setting (when bits RST2 to RST0 are set to 010b) Cycle: $\frac{n + 2}{fBTi}$ Low level width: $\frac{m}{fBTi}$ High level width: $\frac{n + 2 - m}{fBTi}$ m: GiPOj register (j = 1 to 7) setting value, 0000h to FFFFh n: GiPO0 register setting value, 0001h to FFFDh If m ≥ n + 2, the output level is fixed to low
Waveform output start condition (2)	The IFEj (j = 0 to 7) bit in the GiFE register is set to 1 (the function for channel j is enabled)
Waveform output stop condition	The IFEj bit is set to 0 (the function for channel j is disabled)
Interrupt request	When the POijR bit in the intelligent I/O interrupt request register is set to 1 (interrupts requested) by matching the base timer value with the GiPOj register setting (Refer to Figure 10.12)
IIOi_j output pin (or OUTC2_j pin for Group 2) function	Pulse signal output
Selectable functions	• Default value setting This function determines the starting waveform output level • Output level inversion This function inverts the waveform output level and output the inverted signal from the IIOi_j pin (or OUTC2_j pin for Group 2)

Notes:
1. When the INV bit in the GiPOCRj register is set to 1 (output level is inverted), widths low and high are inverted.
2. To use channels shared by time measurement and waveform generation, the FSCj bit in the GiFS register should be set to 0 (waveform generation is selected).

Case 1: Free-running operation (Bits RST2 to RST0 in the GiBCR register are set to 000b)

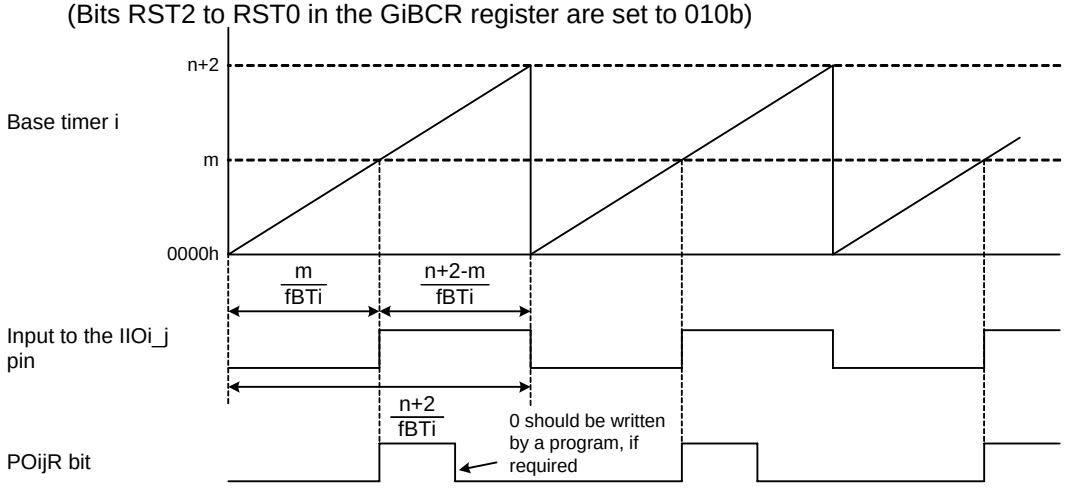


j = 0 to 7
m: GiPOj register setting value (0000h to FFFFh)
POijR bit: Bit in registers IIO0IR to IIO11IR

Notes:
1. Output waveform when the INV bit in the GiPOCRj register is set to 0 (output level is not inverted) and the IVL bit is set to 0 (low level is output as default value).
2. Output waveform when the INV bit is set to 0 (output level is not inverted) and the IVL bit is set to 1 (high level is output as default value).

This figure above applies under the following condition:
- Bits UD1 and UD0 in the GiBCR1 register are set to 00b (increment counting mode)

Case 2: The base timer is reset by matching the base timer value with the GiPO0 register setting (Bits RST2 to RST0 in the GiBCR register are set to 010b)



j = 1 to 7
m: GiPOj register setting value (0000h to FFFFh)
n: GiPO0 register setting value (0001h to FFFDh)
POijR bit: Bit in registers IIO0IR to IIO11IR

This figure above applies under the following conditions:
- The IVL bit in the GiPOCRj register is set to 0 (low level is output as default value) and the INV bit is set to 0 (output is not inverted)
- Bits UD1 and UD0 in the GiBCR1 register are set to 00b (increment counting mode)
- m < n+2

Figure 22.25 Single-phase Waveform Output Mode Operation (i = 0 to 2)

22.3.2 Inverted Waveform Output Mode (for Groups 0 to 2)

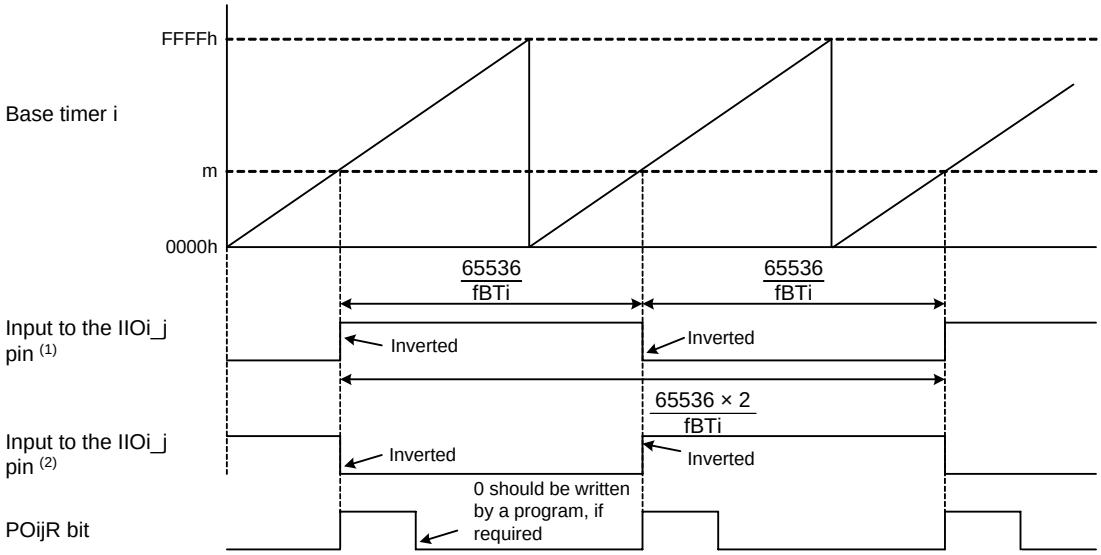
The output level at the IIOi_j pin (or OUTC2_j pin for Group 2) is inverted every time the base timer value matches the GiPOj register setting (i = 0 to 2; j = 0 to 7).
Table 22.8 lists specifications of the inverted waveform output mode. Figure 22.26 shows an example of the inverted waveform output mode operation.

Table 22.8 Inverted Waveform Output Mode Specifications (i = 0 to 2)

Item	Specification
Output waveform	• Free-running operation (when bits RST2 to RST0 in the GiBCR1 register are set to 000b) Cycle: $\frac{65536 \times 2}{fBTi}$ High or low level width: $\frac{65536}{fBTi}$ m: GiPOj register (j = 0 to 7) setting value, 0000h to FFFFh • The base timer is reset by matching the base timer value with the GiPO0 register setting (when bits RST2 to RST0 are set to 010b) Cycle: $\frac{2(n+2)}{fBTi}$ High or low level width: $\frac{n+2}{fBTi}$ n: GiPO0 register setting value, 0001h to FFFDh GiPOj register (j = 1 to 7) setting value, 0000h to FFFFh If the GiPOj register setting ≥ n+2, the output level is not inverted
Waveform output start condition (1)	The IFEj bit in the GiFE register (j = 0 to 7) is set to 1 (the function for channel j is enabled)
Waveform output stop condition	The IFEj bit is set to 0 (the function for channel j is disabled)
Interrupt request	When the POijR bit in the intelligent I/O interrupt request register is set to 1 (interrupts requested) by matching the base timer value with the GiPOj register setting (Refer to Figure 10.12)
IIOi_j output pin (or OUTC2_j pin for Group 2) function	Pulse signal output
Selectable functions	• Default value setting This function determines the starting waveform output level • Output level inversion This function inverts the waveform output level and outputs the inverted signal from the IIOi_j pin (or OUTC2_j pin for Group 2)

Note:
1. To use channels shared by time measurement and waveform generation, the FSCj bit in the GiFS register should be set to 0 (waveform generation is selected).

Case 1: Free-running operation (Bits RST2 to RST0 in the GiBCR1 register are set to 000b)

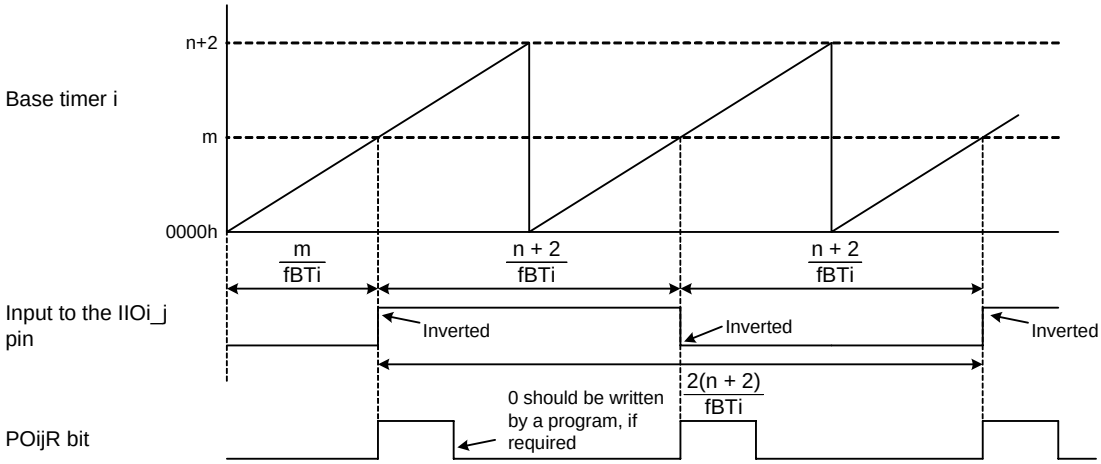


j = 0 to 7
m: GiPOj register setting value (0000h to FFFFh)
POijR bit: Bit in registers IIO0IR to IIO11IR

Notes:
1. Output waveform when the INV bit in the GiPOCRj register is set to 0 (output is not inverted) and the IVL bit is set to 0 (low level is output as default value).
2. Output waveform when the INV bit is set to 0 (output is not inverted) and the IVL bit is set to 1 (high level is output as default value).

This figure above applies under the following condition:
- Bits UD1 and UD0 in the GiBCR1 register are set to 00b (increment counting mode)

Case 2: The base timer is reset by matching the base timer value with the GiPO0 register setting (Bits RST2 to RST0 are set to 010b)



j = 1 to 7
m: GiPOj register setting value (0000h to FFFFh)
n: GiPO0 register setting value (0001h to FFFDh)
POijR bit: Bit in registers IIO0IR to IIO11IR

This figure above applies under the following conditions:
- The IVL bit in the GiPOCRj register is set to 0 (low level is output as default value) and the INV bit is set to 0 (output is not inverted)
- Bits UD1 and UD0 in the GiBCR1 register are set to 00b (increment counting mode)
- m < n+2

Figure 22.26 Inverted Waveform Output Mode Operation (i = 0 to 2)

22.3.3 Set/Reset Waveform Output Mode (SR Waveform Output Mode) (for Groups 0 to 2)

The output level at the IIOi_j pin (or OUTC2_j pin for Group 2) becomes high when the base timer value matches the GiPOj register setting (i = 0 to 2; j = 0, 2, 4, 6). It switches to low when the base timer value matches the GiPOk register setting (k = j + 1) or the base timer reaches 0000h. If the IVL bit in the GiPOCRj register (j = 0 to 7) is set to 1 (high level is output as default value), a high output level is provided when a waveform output starts. If the INV bit is set to 1 (output level is inverted), a waveform with inverted level is output. Refer to Figure 22.27 for details on SR waveform mode operation. Table 22.9 lists specifications of SR waveform output mode.

Table 22.9 SR Waveform Output Mode Specifications (i = 0 to 2)

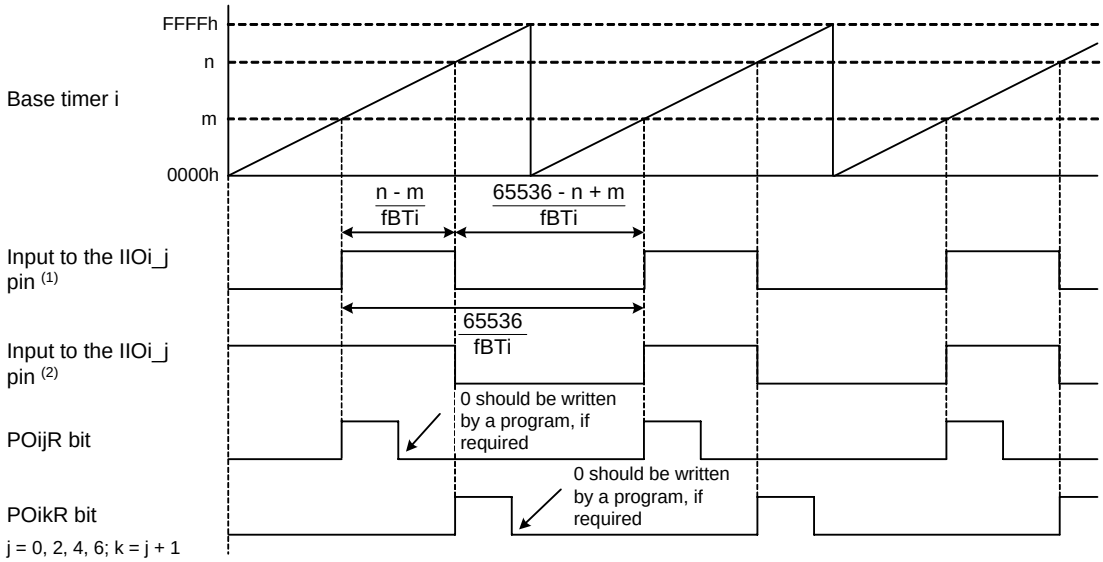
Item	Specification
Output waveform (1)	• Free-running operation (when bits RST2 to RST0 in the GiBCR1 register are set to 000b) (1) m < n High level width: $\frac{n-m}{fBTi}$ Low level width: $\frac{m}{fBTi}$ (2) + $\frac{65536-n}{fBTi}$ (3) (2) m ≥ n High level width: $\frac{65536-m}{fBTi}$ Low level width: $\frac{m}{fBTi}$ m: GiPOj register (j = 0, 2, 4, 6) setting value, 0000h to FFFFh n: GiPOk register (k = j + 1) setting value, 0000h to FFFFh • The base timer is reset by matching the base timer value with the GiPO0 register setting (when bits RST2 to RST0 are set to 010b) (4) (1) m < n < p+2 High level width: $\frac{n+m}{fBTi}$ Low width: $\frac{m}{fBTi}$ (2) + $\frac{p+2-n}{fBTi}$ (3) (2) m < p+2 ≤ n High level width: $\frac{p+2-m}{fBTi}$ Low level width: $\frac{m}{fBTi}$ (3) m ≥ p+2, output level is fixed to low p: GiPO0 register setting value, 0001h to FFFDh m: GiPOj register (j = 2, 4, 6) setting value, 0000h to FFFFh n: GiPOk register (k = j + 1) setting value, 0000h to FFFFh
Waveform output start Condition (5)	The IFEq bit (q = 0 to 7) in the GiFE register is set to 1 (the function for channel q is enabled)
Waveform output stop condition	The IFEq bit is set to 0 (the function for channel q is disabled)

Table 22.9 SR Waveform Output Mode Specifications (i = 0 to 2)

Item	Specification
Interrupt request	When the POijR bit in the intelligent I/O interrupt request register is set to 1 (interrupts requested) by matching the base timer value with the GiPOj register setting. When the POikR bit is set to 1 (interrupts requested) by matching the base timer value with the GiPOk register setting (refer to Figure 10.12)
IIOi_j output pin (or OUTC2_j pin for Group 2) function	Pulse signal output
Selectable functions	• Default value setting This function determines the starting waveform output level • Output level inversion This function inverts the waveform output level and output the inverted signal from the IIOi_j pin (or OUTC2_j pin for Group 2)

Notes:
1. When the INV bit in the GiPOCRj register is set to 1 (output is inverted), widths low and high are inverted.
2. Output period from a base timer reset until when the output level becomes high.
3. Output period from when the output level becomes low until the next base timer reset.
4. When the GiPO0 register resets the base timer, channels 0 and 1 SR waveform generation functions are not available.
5. To use channels shared by time measurement and waveform generation, the FSCj bit in the GiFS register should be set to 0 (waveform generation is selected).

Case 1: Free-running operation (Bits RST2 to RST0 in the GiBCR1 register are set to 000b)

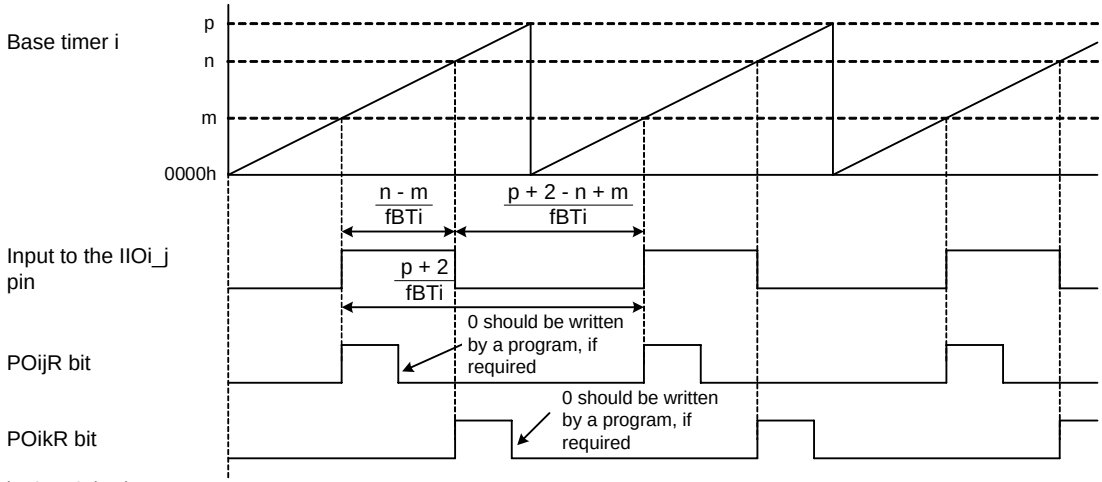

FFFFh
n
m
0000h
Base timer i
n - m / fBTi
65536 - n + m / fBTi
Input to the IIOi_j pin (1)
65536 / fBTi
Input to the IIOi_j pin (2)
0 should be written by a program, if required
POijR bit
0 should be written by a program, if required
POikR bit

j = 0, 2, 4, 6; k = j + 1
m: GiPOj register setting value, 0000h to FFFFh
n: GiPOk register setting value, 0000h to FFFFh
POijR bit, POikR bit: Bits in registers IIO0IR to IIO11IR

Notes:
1. Output waveform when the INV bit in the GiPOCRj register is set to 0 (output is not inverted) and the IVL bit is set to 0 (low level is output as default value).
2. Output waveform when the INV bit is set to 0 (output is not inverted) and the IVL bit is set to 1 (high level is output as default value).

This figure above applies under the following conditions:
- Bits UD1 and UD0 in the GiBCR1 register are set to 00b (increment counting mode)
- m < n

Case 2: The base timer is reset by matching the base timer value with the GiPO0 register setting (Bits RST2 to RST0 in the GiBCR1 register are set to 010b)


p
n
m
0000h
Base timer i
n - m / fBTi
p + 2 - n + m / fBTi
Input to the IIOi_j pin
p + 2 / fBTi
0 should be written by a program, if required
POijR bit
0 should be written by a program, if required
POikR bit

j = 2, 4, 6; k = j + 1
m: GiPOj register setting value, 0000h to FFFFh
n: GiPOk register setting value, 0000h to FFFDh
p: GiPO0 register setting value, 0001h to FFFDh
POijR bit, POikR bit: Bits in registers IIO0IR to IIO11IR

This figure above applies under the following conditions:
- The IVL bit is set to 0 (low level is output as default value) and the INV bit is set to 0 (output is not inverted).
- Bits UD1 and UD0 in the GiBCR1 register are set to 00b (increment counting mode)
- m < n < p + 2

Figure 22.27 SR Waveform Output Mode Operation (i = 0 to 2)

22.3.4 Bit Modulation PWM Output Mode (for Group 2)

In bit modulation PWM output mode, a PWM output has a 16-bit resolution.
Pulses are output in repetitive cycles, each cycle consisting of span t repeated 1024 times. The span t has a cycle of $\frac{64}{fBT2}$. The six upper bits in the G2POj register (j = 0 to 7) determine the base low width.
The ten lower bits determine the number of span t, within a cycle, in which low width is extended by the minimum resolution bit width, that is, 1 clock cycle.
If the INV bit is set to 1 (output level is inverted), the waveform with inverted level is output.
Table 22.10 lists specifications of bit modulation PWM output mode. Table 22.11 lists the number of modulated spans and span ts to be extended with the minimum resolution bit width. Figure 22.28 shows an example of bit modulation PWM output mode operation.

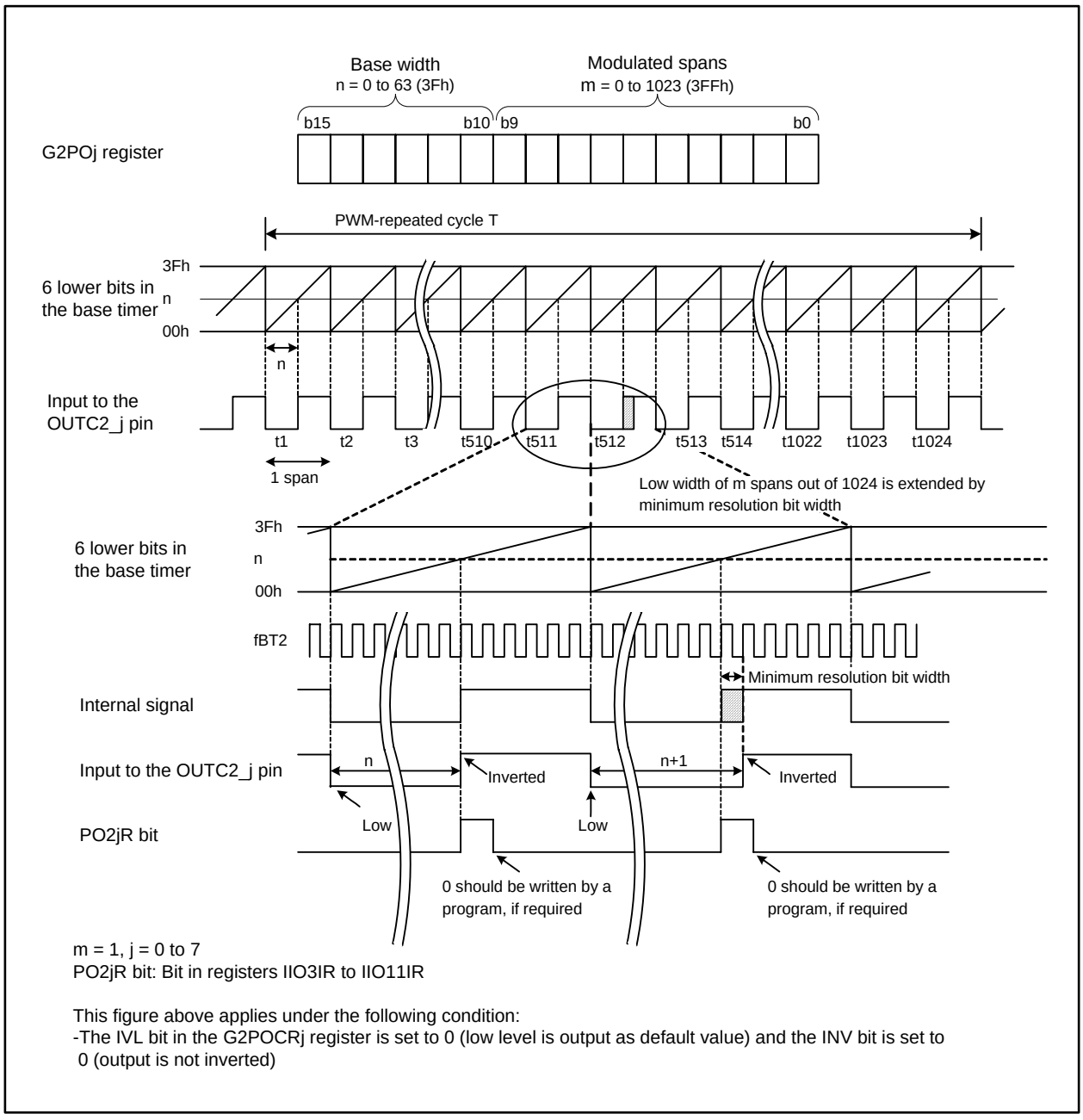
Table 22.10 Bit Modulation PWM Output Mode Specifications

Item	Specification
Output waveform (1, 2)	PWM-repeated cycle T: $\frac{65536}{fBT2}$ (= $\frac{64}{fBT2}$ × 1024) Cycle of span t: $\frac{64}{fBT2}$ Low width: $\frac{n+1}{fBT2}$ of m spans $\frac{n}{fBT2}$ of (1024-m) spans Mean low width: $\frac{1}{fBT2}$ × (n + $\frac{m}{1024}$) n: G2POj register (j = 0 to 7) setting value (6 upper bits), 00h to 3Fh m: G2POj register (j = 0 to 7) setting value (10 lower bits), 000h to 3FFh
Waveform output start condition	The IFEj bit in the G2FE register (j = 0 to 7) is set to 1 (the function for channel j is enabled)
Waveform output stop condition	The IFEj bit is set to 0 (the function for channel j is disabled)
Interrupt request	When the PO2jR bit in the interrupt request register is set to 1 (interrupts requested) by matching the 6 lower bits of the base timer value with the 6 upper bits of the G2POj register setting (Refer to Figure 10.12)
OUTC2_j pin function	Pulse signal output pin
Selectable functions	• Default value setting This function determines the starting waveform output level • Output level inversion This function inverts the waveform output level and output the inverted signal from the OUTC2_j pin

Notes:
1. Bits RST2 and RST0 in the G2BCR1 register should be set to 000b to use bit modulation PWM output mode.
2. When the INV bit in the G2POCRj register is set to 1 (output level is inverted), widths low and high are inverted.

Table 22.11 Number of Modulated Spans and Span t Extended Minimum Resolution Bit Width

Modulated Spans	Span ts to be Extended with Minimum Resolution Bit Width
00 0000 0000b	none
00 0000 0001b	t512
00 0000 0010b	t256 and t768
00 0000 0100b	t128, t384, t640, and t896
00 0000 1000b	t64, t192, t320, t448, t576, t704, t832, and t960
:	:
10 0000 0000b	t1, t3, t5, t7, ••• t1019, t1021, and t1023



m = 1, j = 0 to 7
PO2jR bit: Bit in registers IIO3IR to IIO11IR

This figure above applies under the following condition:
-The IVL bit in the G2POCRj register is set to 0 (low level is output as default value) and the INV bit is set to 0 (output is not inverted)

Figure 22.28 Bit Modulation PWM Output Mode Operation

22.3.5 Real-time Port Output Mode (RTP Output Mode) (for Group 2)

The OUTC2_j pin (j = 0 to 7) outputs the G2RTP register setting value in one-bit units when the base timer value matches the G2POj register setting. Table 22.12 lists specifications of RTP output mode. Figure 22.29 shows a block diagram of RTP output and Figure 22.30 shows an example of RTP output mode operation.

Table 22.12 RTP Output Mode Specifications

Item	Specification
Waveform output start condition	The IFEj bit (j = 0 to 7) in the G2FE register is set to 1 (the function for channel j is enabled)
Waveform output stop condition	The IFEj bit is set to 0 (the function for channel j is disabled)
Interrupt request	When the PO2jR bit in the interrupt request register is set to 1 (interrupts requested) by matching the base timer value with the G2POj register setting (0000h to FFFFh (1)) (Refer to Figure 10.12)
OUTC2_j pin function	RTP output pin
Selectable functions	• Default value setting This function determines the starting waveform output level • Output level inversion This function inverts the waveform output level and output the inverted signal from the OUTC2_j pin

Note:
1. The G2PO0 register should be set to between 0001h and FFFDh to set the base timer value to 0000h (bits RST2 to RST0 are set to 010b) when the base timer value matches the G2PO0 register setting.

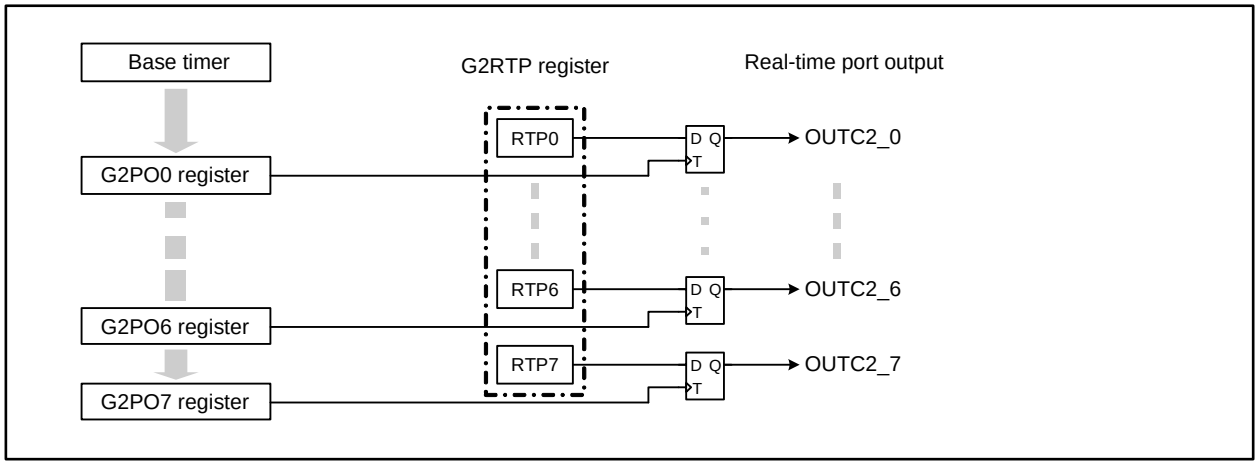
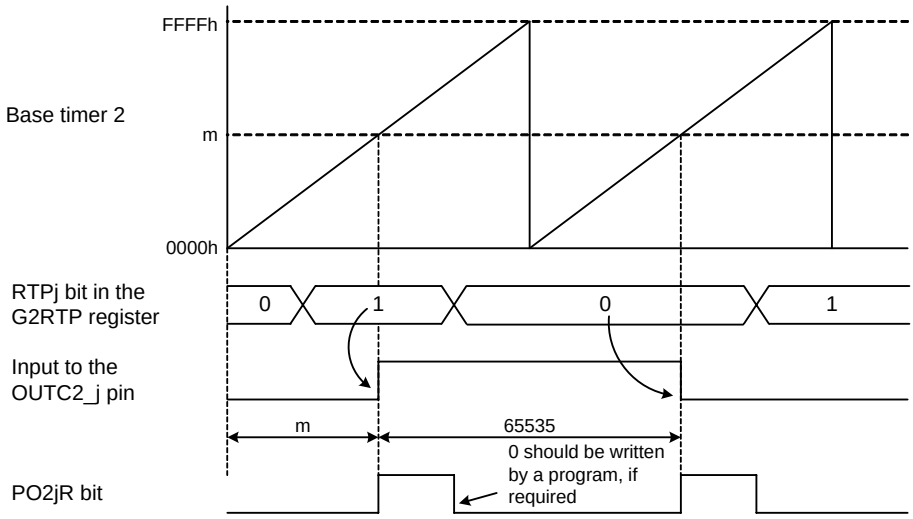


Figure 22.29 RTP Output Block Diagram

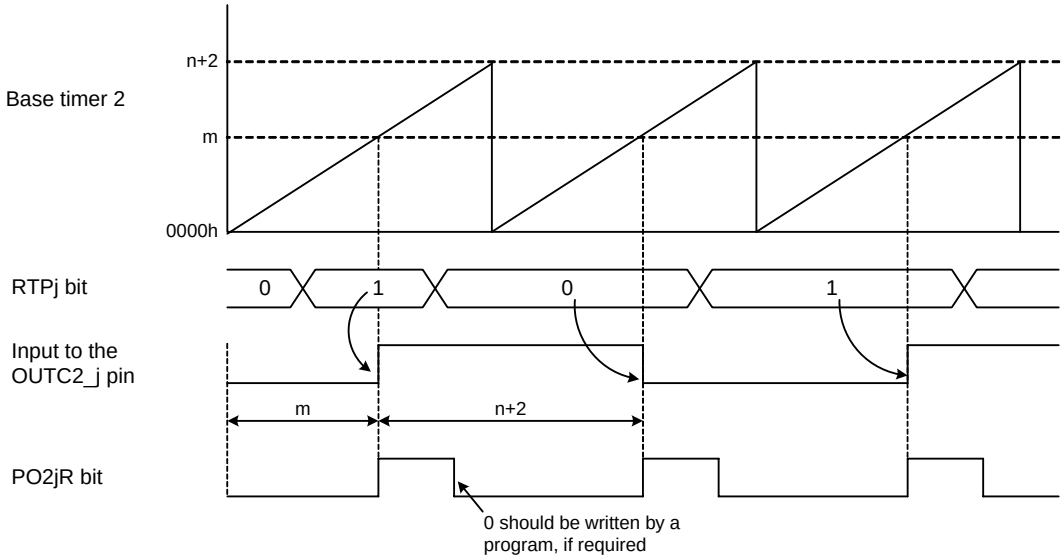
Case 1: Free-running operation (Bits RST2 to RST0 in the G2BCR1 register are set to 000b)



j = 0 to 7
m: G2POj register setting value, 0000h to FFFFh
PO2jR bit: Bit in registers IIO03R to IIO11IR

This figure above applies under the following conditions:
-The IVL bit in the G2POCRj register is set to 0 (low level is output as default value) and the INV bit is set to "0" (output is not inverted).
-Bits RST2 to RST0 in the G2BCR1 register are set to 000b (base timer is not reset)

Case 2: The base timer is reset by matching the base timer value with the G2PO0 register setting (Bits RST2 to RST0 are set to 010b)



j = 1 to 7
m: G2POj register setting value, 0000h to FFFFh
n: G2POj register setting value, 0001h to FFFDh
PO2jR bit: Bit in registers IIO03R to IIO11IR

This figure above applies under the following conditions:
-The IVL bit in the G2POCRj register is set to 0 (low level is output as default value) and the INV bit is set to 0 (output level is not inverted)
-m < n + 2

Figure 22.30 RTP Output Mode Operation

22.3.6 Parallel Real-time Port Output Mode (RTP Output Mode) (for Group 2)

The OUTC2_j pin (j = 0 to 7) outputs all the G2RTP register setting values in one-byte units when the base timer value matches the G2POj register setting. Table 22.13 lists specifications of parallel RTP output mode. Figure 22.7 shows the G2BCR1 register. Figure 22.31 shows a block diagram of parallel RTP output and Figure 22.32 shows an example of parallel RTP output mode operation.

Table 22.13 Parallel RTP Output Mode Specifications

Item	Specification
Waveform output start condition	The IFEj bit (j = 0 to 7) in the G2FE register is set to 1 (the function for channel j is enabled)
Waveform output stop Condition	The IFEj bit is set to 0 (the function for channel j is disabled)
Interrupt request	When the PO2jR bit in the interrupt request register is set to 1 (interrupts requested) by matching the base timer value with the G2POj register setting (0000h to FFFFh (1)) (Refer to Figure 10.12)
OUTC2_j pin function	RTP output pin
Selectable functions	• Default value setting This function determines the starting waveform output level • Output level inversion This function inverts the waveform output level and output the inverted signal from the OUTC2_j pin

Note:
1. The G2PO0 register should be set to between 0001h and FFFDh to set the base timer value to 0000h (bits RST2 to RST0 are set to 010b) when the base timer value matches the G2PO0 register setting.

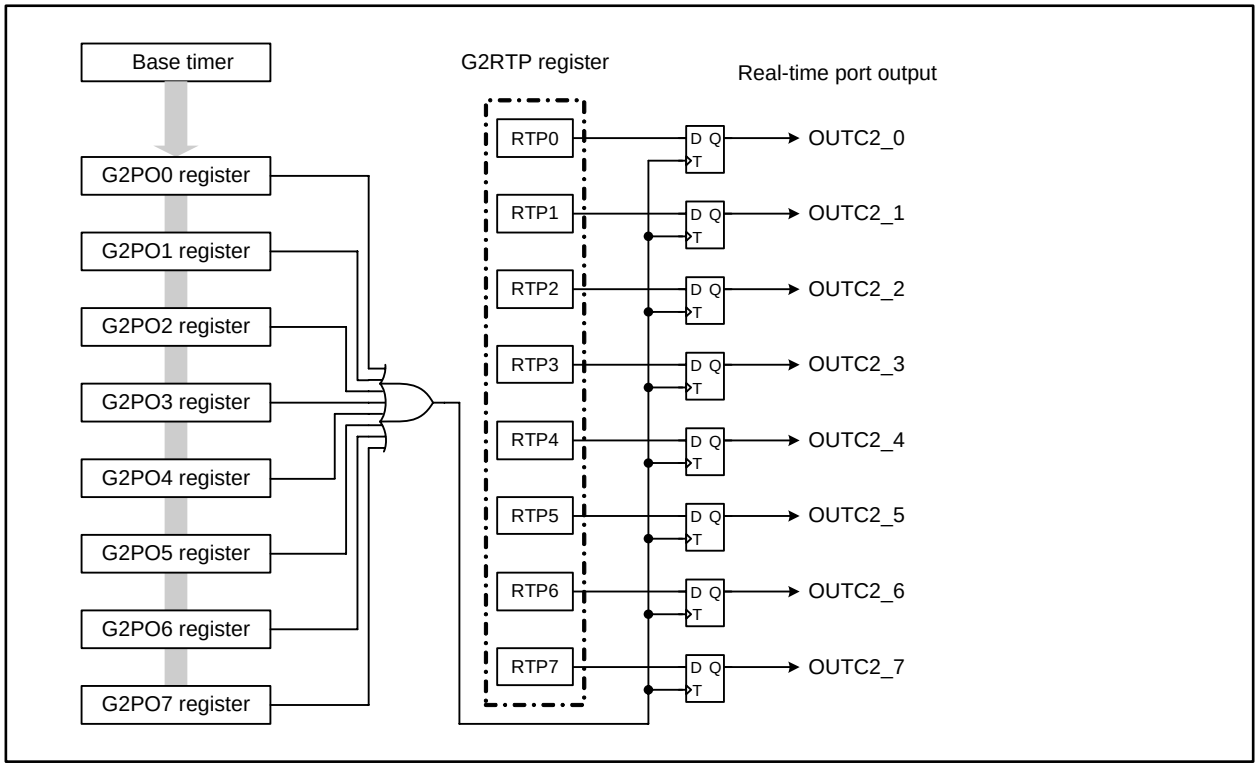
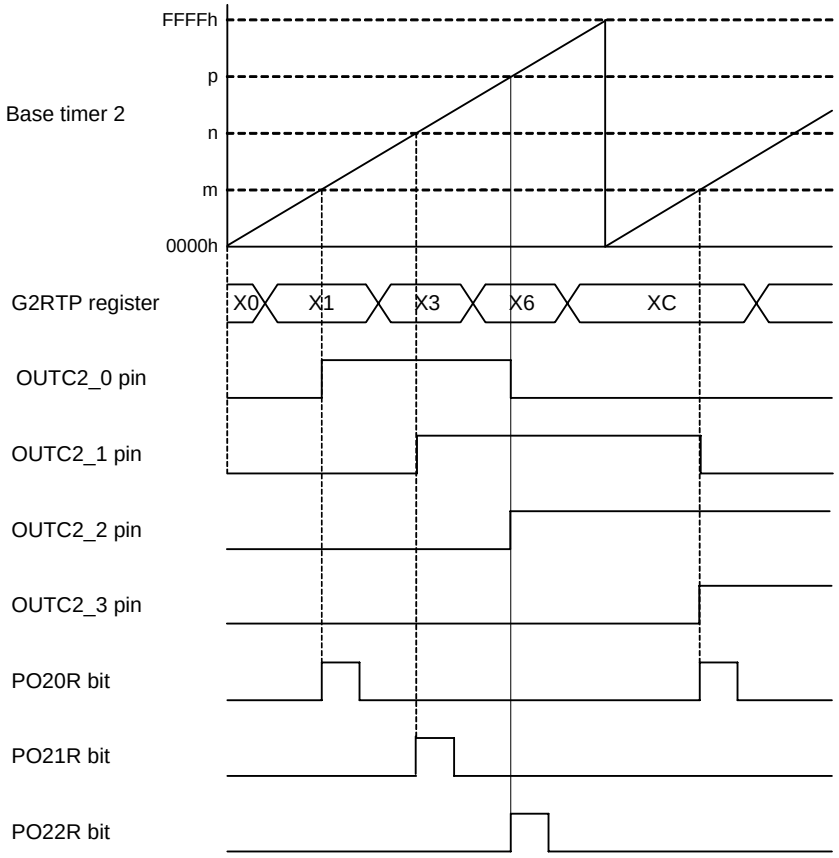


Figure 22.31 Parallel RTP Output Mode Block Diagram

Case 1: Free-running operation



m: G2PO0 register setting value, 0000h to FFFFh
n: G2PO1 register setting value, 0000h to FFFFh
p: G2PO2 register setting value, 0000h to FFFFh
Bits PO20R, PO21R, and PO22R: Bits in registers IIO3IR to IIO11IR

This figure above applies under the following conditions:
-The IVL bit in the G2POCRj register is set to 0 (low level is output as default value) and the IVL bit is set to 0 (output level is not inverted)
-Bits RST2 to RST0 in the G2BCR1 register are set to 000b (base timer is not reset)
-m < n < p

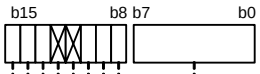
Figure 22.32 Parallel RTP Output Mode Operation

22.4 Group 2 Serial Interface

Two 8-bit shift registers and waveform generation enable the serial interface function. In group 2 of the intelligent I/O, the variable synchronous serial interface and IEBus (optional (1)) are available.
Figure 22.33 to Figure 22.40 show associated registers.

Note:
1. Contact a Renesas Electronics sales office to use the optional features.

Group 2 SI/O Transmit Buffer Register



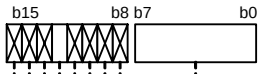
Symbol	Address	Reset Value
G2TB	016Dh-016Ch	Undefined

Bit Symbol	Bit Name	Function	RW
— (b7-b0)	Transmit Buffer	Data transmitted	WO
SZ0	Transmit/Receive Character Length Select Bit	b10 b9 b8 0 0 0 : 8 bits 0 0 1 : 1 bit 0 1 0 : 2 bits 0 1 1 : 3 bits 1 0 0 : 4 bits 1 0 1 : 5 bits 1 1 0 : 6 bits 1 1 1 : 7 bits	RW
SZ1			RW
SZ2			RW
— (b12-b11)	No register bits; should be written with 0 and read as undefined value		—
A	ACK Function Select Bit	0: Do not add the ACK bit 1: Add the ACK bit after last transmit bit	RW
PC	Parity Calculation Continuing Bit	0: Add the parity bit after this transmit data 1: Carry over a parity to the data to be transmitted (1)	RW
P	Parity Select Bit	0: No parity 1: Parity (even parity only)	RW

Note:
1. The PC bit should be set to 1 after the P bit is set to 0.

Figure 22.33 G2TB Register

Group 2 SI/O Receive Buffer Register



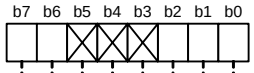
Symbol	Address	Reset Value
G2RB	016Fh-016Eh	Undefined

Bit Symbol	Bit Name	Function	RW
— (b7-b0)	Receive Buffer	Data received	RO
— (b11-b8)	No register bits; should be written with 0 and read as undefined value		—
OER	Overrun Error Flag (1)	0: No overrun error 1: Overrun error	RO
— (b15-b13)	No register bits; should be written with 0 and read as undefined value		—

Note:
1. The OER bit is set to 0 when bits GMD1 and GMD0 in the G2MR register are set to 00b (the communication block is reset) or the RE bit in the G2CR register is set to 0 (reception disabled).

Figure 22.34 G2RB Register

Group 2 Serial Interface Mode Register



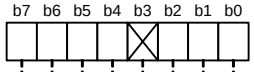
Symbol	Address	Reset Value
G2MR	016Ah	00XX X000b

Bit Symbol	Bit Name	Function	RW
GMD0	Serial Interface Mode Select Bit	b1 b0 0 0 : The communication block is reset (the OER bit is set to 0) (1) 0 1 : Variable synchronous serial interface mode (2) 1 0 : IEBus mode (2) 1 1 : Do not use this combination	RW
GMD1			RW
CKDIR	Internal/External Clock Select Bit	0: Internal clock 1: External clock	RW
— (b5-b3)	No register bits; should be written with 0 and read as undefined value		—
UFORM	Bit Order Select Bit	0: LSB first 1: MSB first	RW
IRS	Transmit Interrupt Source Select Bit	0: Transmit buffer is empty 1: Transmission is completed	RW

Notes:
1. One base timer clock or more is required after bits GMD1 and GMD0 are set to 00b.
2. Bits GMD1 and GMD0 should be set to 01b or 10b while the base timer clock is stopped.

Figure 22.35 G2MR Register

Group 2 Serial Interface Control Register



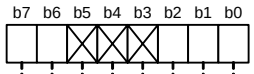
Symbol	Address	Reset Value
G2CR	016Bh	0000 X110b

Bit Symbol	Bit Name	Function	RW
TE	Transmit Enable Bit	0: Transmission disabled 1: Transmission enabled	RW
TXEPT	Transmit Shift Register Empty Flag	0: Data in the transmit shift register (during transmission) 1: No data in the transmit shift register (transmission completed)	RO
TI	Transmit Buffer Empty Flag	0: Data held in the G2TB register 1: No data held in the G2TB register	RO
— (b3)	No register bit; should be written with 0 and read as undefined value		—
RE	Receive Enable Bit (1)	0: Reception disabled 1: Reception enabled	RW
RI	Receive Complete Flag	0: No data held in the G2RB register 1: Data held in the G2RB register	RO
OPOL	ISTXD2 Output Polarity Switching Bit	0: Not inverted 1: Inverted	RW
IPOL	ISRXD2 Input Polarity Switching Bit (1)	0: Not inverted 1: Inverted	RW

Note:
1. The group 2 base timer may be reset when these bits are rewritten. To avoid unexpected resets, the RST2 bit in the G2BCR1 register should be set to 0 (base timer is not reset by a reset request from the serial interface).

Figure 22.36 G2CR Register

Group 2 IEBus Control Register



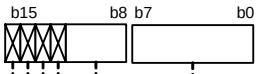
Symbol	Address	Reset Value
IECR	0172h	00XX X000b

Bit Symbol	Bit Name	Function	RW
IEB	IEBus Enable Bit (1)	0: IEBus disabled (2) 1: IEBus enabled	RW
IETS	IEBus Transmit Start Request Bit	0: Transmission completed 1: Transmission started	RW
IEBBS	IEBus Bus Busy Flag	0: Idle state 1: Busy state (start condition detected)	RO
— (b5-b3)	No register bits; should be written with 0 and read as undefined value		—
DF	Digital Filter Select Bit	0: No digital filter 1: Use the digital filter	RW
IEM	IEBus Mode Select Bit	0: Mode 1 1: Mode 2	RW

Notes:
1. The IEB bit should be rewritten while the base timer clock is stopped.
2. One fBT2 cycle or more is required after the IEB bit is set to 0. To set this bit to 1, bits BCK1 and BCK0 in the G2BCR0 register should be set to 00b (clock is stopped).

Figure 22.37 IECR Register

Group 2 IEBus Address Register

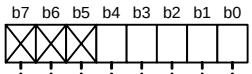


Symbol	Address	Reset Value
IEAR	0171h-0170h	Undefined

Function	RW
Address data	RW
Address data	RW
No register bits; should be written with 0 and read as undefined value	—

Figure 22.38 IEAR Register

Group 2 IEBus Transmit Interrupt Source Detect Register



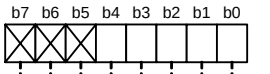
Symbol	Address	Reset Value
IETIF	0173h	XXX0 0000b

Bit Symbol	Bit Name	Function	RW
IETNF	Normal Completion Flag	0: Transmission is completed in error 1: Transmission is successfully completed (1)	RW
IEACK	ACK Error Flag	0: No error detected 1: Error detected (1)	RW
IETMB	Maximum Transmit Byte Error Flag	0: No error detected 1: Error detected (1)	RW
IETT	Timing Error Flag	0: No error detected 1: Error detected (1)	RW
IEABL	Arbitration Lost Flag	0: No error detected 1: Error detected (1)	RW
— (b7-b5)	No register bits; should be written with 0 and read as undefined value		—

Note:
1. This bit can be set to 0 by a program but cannot be set to 1. To set this bit to 0, the IEB bit in the IECR register should be set to 0 (IEBus disabled).

Figure 22.39 IETIF Register

Group 2 IEBus Receive Interrupt Source Detect Register



Symbol	Address	Reset Value
IERIF	0174h	XXX0 0000b

Bit Symbol	Bit Name	Function	RW
IERNF	Normal Completion Flag	0: Reception is completed in error 1: Reception is successfully completed (1)	RW
IEPAR	Parity Error Flag	0: No error detected 1: Error detected (1)	RW
IERMB	Maximum Receive byte Error Flag	0: No error detected 1: Error detected (1)	RW
IERT	Timing Error Flag	0: No error detected 1: Error detected (1)	RW
IERETC	Error by Other Sources Flag	0: No error detected 1: Error detected (1)	RW
— (b7-b5)	No register bits; should be written with 0 and read as undefined value		—

Note:
1. This bit can be set to 0 by a program but cannot be set to 1. To set this bit to 0, the IEB bit in the IECR register should be set to 0 (IEBus disabled).

Figure 22.40 IERIF Register

22.4.1 Variable Synchronous Serial Interface Mode (for Group 2)

This mode allows 1-bit to 8-bit data transmission/reception synchronized with the transmit/receive clock. The character length is selectable from 1 to 8 bits. Table 22.14 lists specifications of the group 2 variable synchronous serial interface mode and Table 22.15 lists its settings. Figure 22.41 shows an operation example of data transmission/reception.

Table 22.14 Group 2 Variable Synchronous Serial Interface Mode Specifications

Item	Specification
Data format	1- to 8-bit character length
Transmit/receive clock	• The CKDIR bit in the G2MR register is set to 0 (internal clock selected): $\frac{fBT2}{2(n+2)}$ n: G2PO0 register setting value, 0000h to FFFFh (1) The bit rate is set using the G2PO0 register. The clock is generated in the inverted waveform output mode of the channel 2 waveform generation • The CKDIR bit is set to 1 (external clock selected): input into the ISCLK2 pin (2)
Transmit start conditions	The conditions for starting data transmission are as follows: • The TE bit in the G2CR register is set to 1 (transmission enabled) • The TI bit in the G2CR register is set to 0 (data held in the G2TB register)
Receive start conditions	The conditions for starting data reception are as follows: • The RE bit in the G2CR register is set to 1 (reception enabled) • The TE bit in the G2CR register is set to 1 (transmission enabled) • The TI bit in the G2CR register is set to 0 (data held in the G2TB register)
Interrupt request	In transmit interrupt, either of the following conditions is selected to set the SIO2TR bit in the IIO6IR register to 1 (interrupts requested) (Refer to Figure 10.12): • The IRS bit in the G2MR register is set to 0 (transmit buffer in the G2TB register is empty): when data is transferred from the G2TB register to the transmit shift register (when the transmission has started) • The IRS bit is set to 1 (transmission is completed): when data transmission from the transmit shift register is completed In receive interrupt, When data is transferred from the receive shift register to the G2RB register (when the reception is completed), the SIO2PR bit in the IIO5IR register is set to 1 (interrupts requested) (Refer to Figure 10.12)
Error detection	Overrun error (3) This error occurs when the last bit of the next data has been received before reading the G2RB register
Selectable functions	• Bit order selection Selectable either LSB first or MSB first • ISTXD2 and ISRXD2 I/O polarity Output level from the ISTXD2 pin and input level to the ISRXD2 pin can be respectively inverted • Character length for data transmission/reception Selectable a character length from 1 to 8 bits

Notes:
1. When using the serial interface, set 1 or above to the G2PO0 register.
2. The highest transmit/receive clock frequency should be fBT2 divided by 20.
3. If an overrun error occurs, the G2RB register is undefined.

Table 22.15 Register Settings in Group2 Variable Synchronous Serial Interface Mode

Register	Bits	Function
G2BCR0	BCK1 and BCK0	Set the bits to 11b
	DIV4 to DIV0	Select a divide ratio of count source
	IT	Set the bit to 0
G2BCR1	7 to 0	Set the bits to 0001 0010b
G2POCR0	7 to 0	Set the bits to 0000 0111b
G2POCR1	7 to 0	Set the bits to 0000 0111b
G2POCR2	7 to 0	Set the bits to 0000 0010b
G2PO0	15 to 0	Set a comparative value for waveform generation $\frac{fBT2}{2 \times (\text{setting value} + 2)}$ = transmit/receive clock frequency
G2PO2	15 to 0	Set to a value smaller than that in the G2PO0 register setting
G2FE	IFE2 to IFE0	Set the bits to 111b
G2MR	GMD1 and GMD0	Set the bits to 01b
	CKDIR	Select either the internal clock or the external clock
	UFORM	Select either LSB first or MSB first
	IRS	Select a source for transmit interrupt
G2CR	TE	Set the bit to 1 to enable data transmission/reception
	TXEPT	Transmit shift register empty flag
	TI	Transmit buffer empty flag
	RE	Set the bit to 1 to enable data reception
	RI	Receive complete flag
	OPOL	Select if the output level at the ISTXD2 pin is inverted (usually set the bit to 0)
	IPOL	Select if the input level at the ISRXD2 pin is inverted (usually set the bit to 0)
G2TB	15 to 0	Set the data to be transmitted/received and its character length
G2RB	15 to 0	Store received data and error flag

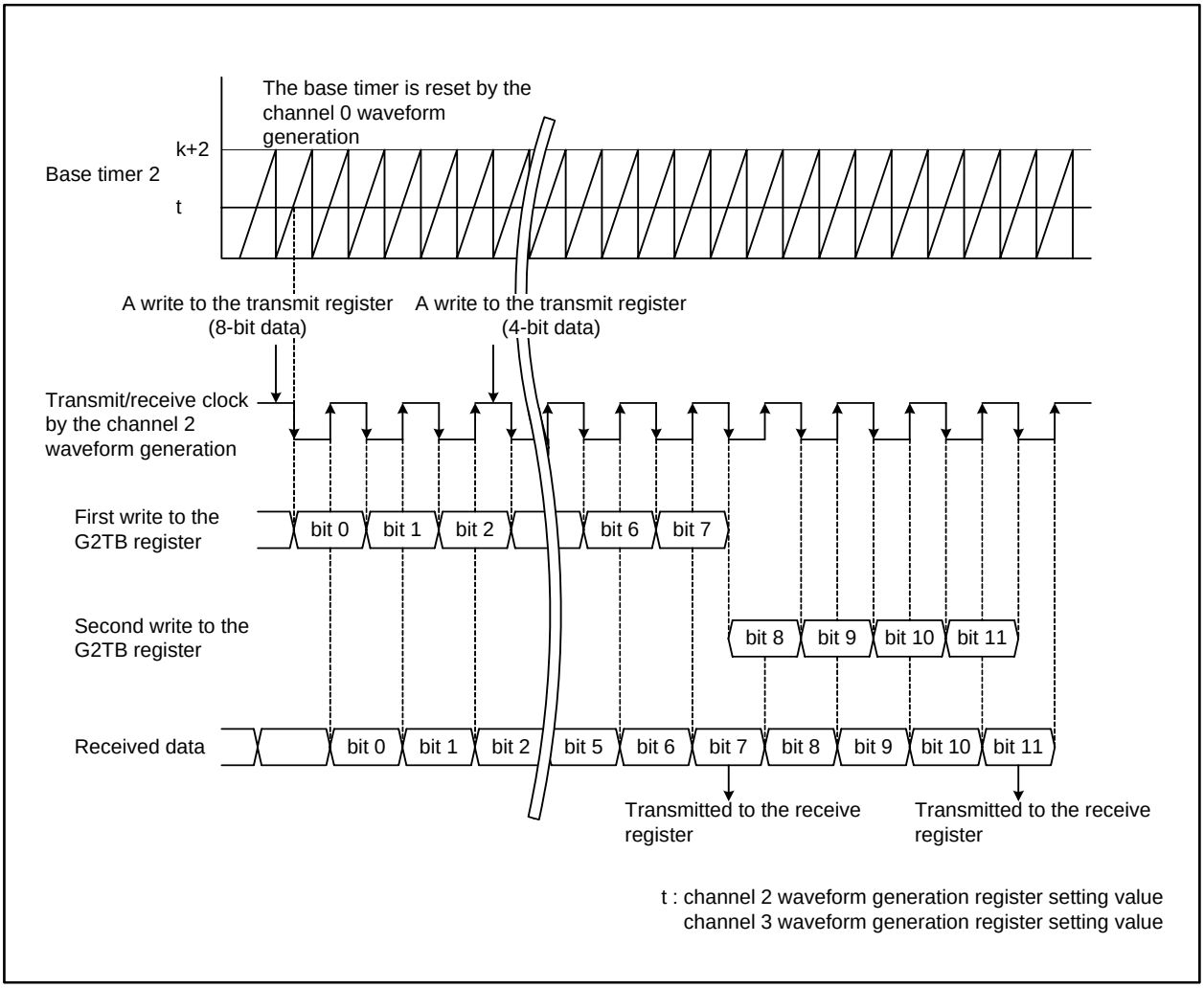


Figure 22.41 Group 2 Variable Synchronous Serial Interface Mode Transmit/Receive Operation

23. Multi-master I2C-bus Interface

The multi-master I2C-bus interface (MMI2C) is capable of serial, bi-directional data transfer in the I2C-bus data transmit and receive format. It contains an arbitration lost detector and a clock synchronization function. Table 23.1 lists specifications of the multi-master I2C-bus interface. Table 23.2 lists detectors of the multi-master I2C-bus interface. Figure 23.1 shows a block diagram of the multi-master I2C-bus interface.

Table 23.1 Multi-master I2C-bus Interface Specifications

Item	Specification
Data format	Compliant with the I2C-bus specification • 7-bit addressing format • Fast-mode • Standard-mode
Master/Slave device	Selectable
I/O pins	Serial data line: MSDA (SDA) Serial clock line: MSCL (SCL)
Transmit/Receive clock	16.1 to 400 kbps (ϕIIC = 4 MHz) ϕIIC: I2C-bus system clock
Transmit/Receive modes	Compliant with the I2C-bus specification • Master-transmit mode • Master-receive mode • Slave-transmit mode • Slave-receive mode
Interrupt request sources	• Six I2C-bus interface interrupts: Successful transmit, successful receive, slave address match detection, general call address detection, STOP condition detection, and timeout detection • Two I2C-bus line interrupts: Rising or falling edge of pins MSDA and MSCL
Selectable functions	• Timeout detector This function detects that the MSCL pin level is held high for longer than the specified time while the bus is busy • Free data format selector This function selects the free data format to generate an interrupt request, regardless of the slave address value, when the first byte is received

Table 23.2 Detectors of Multi-master I2C-bus Interface

Item	Specification
Slave address match detector	In slave-receive mode, this detects whether the address sent from the master device matches the slave address. When they match, an ACK is automatically sent. When they do not, a NACK is automatically sent and communication is stopped
General call address detector	This detects a general call address when in slave-receive mode
Arbitration lost detector	This detects an arbitration lost and then immediately stops output to the MSDA pin.
Bus busy detector	This detects that the bus is busy, and sets/resets the BBSY bit

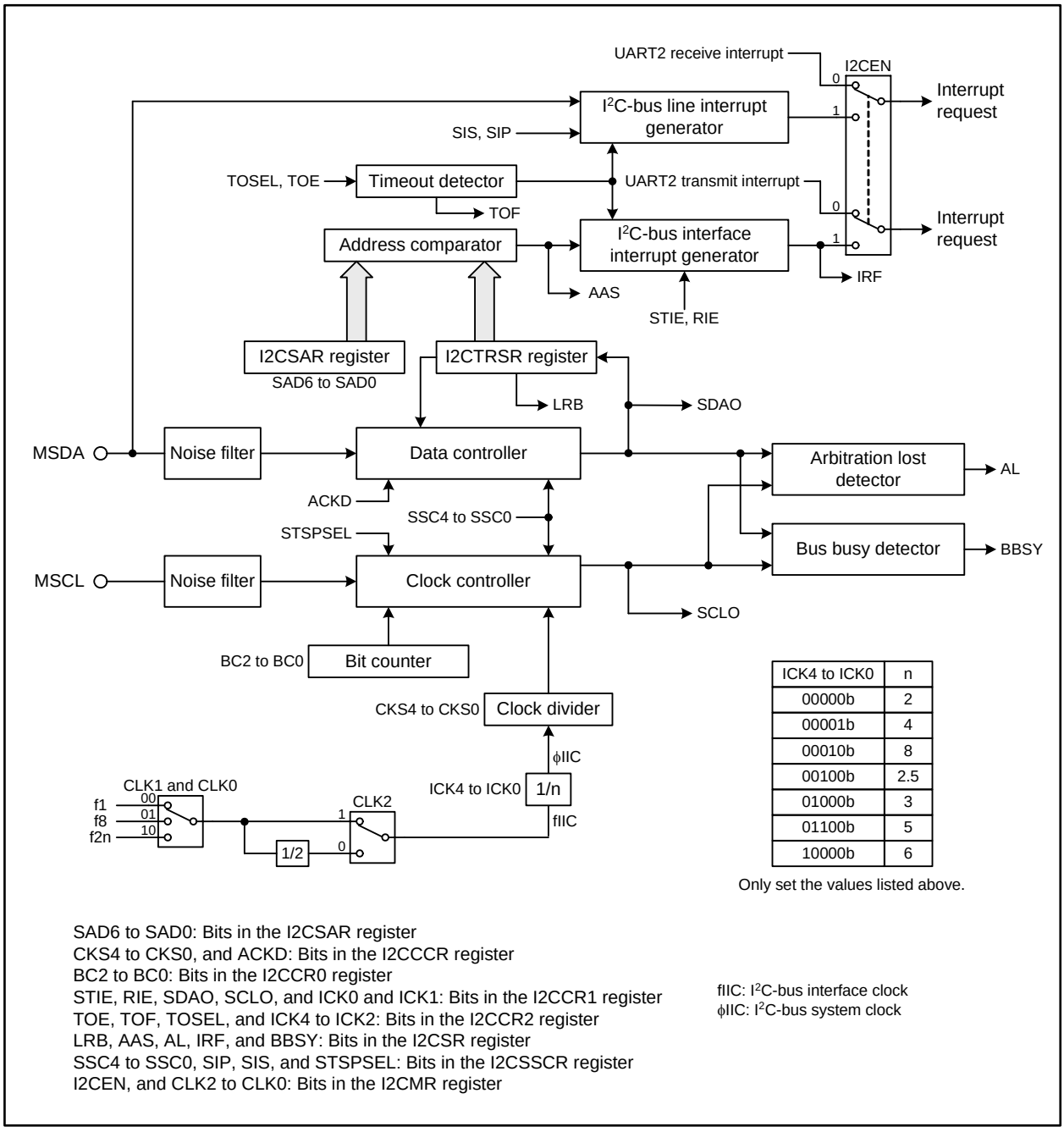


Figure 23.1 Multi-master I2C-bus Interface Block Diagram

23.1 Multi-master I2C-bus Interface-associated Registers

23.1.1 I2C-bus Transmit/Receive Shift Register (I2CTRSR)

I2C-bus Transmit/Receive Shift Register (1, 2, 3)

b7 b0

Symbol	Address	Reset Value
I2CTRSR	044400h	Undefined

Function	RW
Set transmit data in transmit mode. In receive mode, write dummy data before receiving data, and read received data after an interrupt is generated. The data being shifted can be read from this register, regardless of whether it is transmitting or receiving	RW

Notes:
1. This register is write enabled when the ICE bit in the I2CCR0 register is 1 (I2C-bus interface enabled).
2. This register is used for transmitting and receiving. Read the received data before using this register for transmitting.
3. When data is written to this register, bits BC2 to BC0 in the I2CCR0 register become 000b, and bits LRB, AAS, and AL in the I2CSR register become 0.

Figure 23.2 I2CTRSR Register

The I2CTRSR register is an 8-bit shift register where received data is stored and transmit data is written. When transmit data is written to this register, the data is synchronized with the SCL clock and shifted out in descending order from bit 7. Every time a bit is shifted out, the data is shifted to the left by 1 bit. During a receive operation, the data is synchronized with the SCL clock and stored in order starting from bit 0. 1 bit of data is shifted (to the left) for every bit that is input. Figure 23.3 shows the timing when the received data is stored to the I2CTRSR register.

The I2CTRSR register is write enabled when the ICE bit in the I2CCR0 register is 1 (I2C-bus interface enabled). When the ICE bit is 1 and the MST bit in the I2CSR register is 1 (master mode), writing data to the I2CTRSR register resets the bit counter and the SCL clock is output.

Write to the I2CTRSR register when a START condition is generated or the MSCL pin is low. The register can always be read.

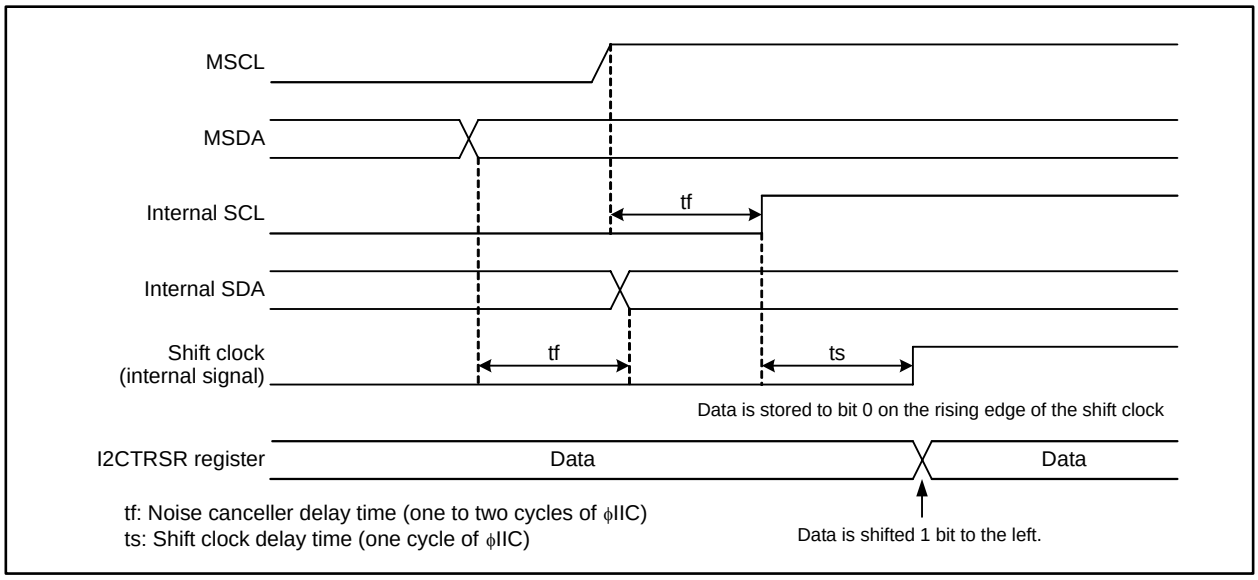


Figure 23.3 Received Data Storing Timing to the I2CTRSR Register

23.1.2 I2C-bus Slave Address Register (I2CSAR)

I2C-bus Slave Address Register

b7	b6	b5	b4	b3	b2	b1	b0
							0

Symbol: I2CSAR
Address: 044402h
Reset Value: 00h

Bit Symbol	Bit Name	Function	RW
— (b0)	Reserved	Should be written with 0	RW
SAD0	Slave Address	The slave address must differ from other slave addresses of slave devices connected to the I2C-bus. In slave mode, the device becomes a slave device when the upper 7 bits sent in the first frame after the START condition match bits SAD6 to SAD0	RW
SAD1			RW
SAD2			RW
SAD3			RW
SAD4			RW
SAD5			RW
SAD6			RW

Figure 23.4 I2CSAR Register

The I2CSAR register stores a slave address to automatically recognize itself as a slave device. When the received address matches the slave address, the device operates as a slave device.

23.1.2.1 Bits SAD6 to SAD0

Bits SAD6 to SAD0 store a slave address. When the addressing format is enabled, the received 7-bit address and the slave address set in bits SAD6 to SAD0 are compared. When a match is detected, the device operates as a slave device.

23.1.3 I2C-bus Control Register 0 (I2CCR0)

I2C-bus Control Register 0

b7	b6	b5	b4	b3	b2	b1	b0
0		0					

Symbol: I2CCR0 — Address: 044403h — Reset Value: 0000 0000b

Bit Symbol	Bit Name	Function	RW
BC0	Transmit/Receive Bit Length Setting Bit (1)	b2 b1 b0 0 0 0 : 8 bits 0 0 1 : 7 bits 0 1 0 : 6 bits 0 1 1 : 5 bits 1 0 0 : 4 bits 1 0 1 : 3 bits 1 1 0 : 2 bits 1 1 1 : 1 bit	RW
BC1			RW
BC2			RW
ICE	I2C-bus Interface Enable Bit	0: I2C-bus interface disabled 1: I2C-bus interface enabled	RW
DFS	Data Format Select Bit	0: Addressing format 1: Free data format	RW
— (b5)	Reserved	Should be written with 0	RW
RST	I2C-bus Interface Reset Bit	Writing 1 to this bit resets the I2C-bus interface circuit	RW
— (b7)	Reserved	Should be written with 0	RW

Note:
1. These bits automatically become 000b in the following cases:
   - When a START or STOP condition is detected
   - When data transmission is completed
   - When data reception is completed

Figure 23.5 I2CCR0 Register

The I2CCR0 register controls data communication format.

23.1.3.1 Bits BC2 to BC0

Bits BC2 to BC0 set the data bit length to be sent or received next. When the data bit length set with bits BC2 to BC0 (acknowledge clock pulse is included in the number when the ACKCLK bit in the I2CCCR register is 1) is sent or received, an I2C-bus interface interrupt request is generated. Consequently, bits BC2 to BC0 become 000b. Note that these bits also become 000b when a START condition is detected. Address data is sent or received in 8 bits regardless of their setting.

23.1.3.2 ICE Bit

The ICE bit enables the I2C-bus interface. Set this bit to 1 to enable the I2C-bus interface and 0 to disable it. When this bit is 0, pins MSDA and MSCL are fixed high (these pins are high-impedance when the corresponding NOD bits in registers P7_0S and P7_1S are 1), therefore the I2C-bus interface cannot be used.

When the ICE bit is set to 0, the following occurs:

- Bits ADZ, AAS, AL, BBSY, TRS, and MST in the I2CSR register become 0, and the IRF bit becomes 1.
- Writing to the I2CTRSR register is disabled.
- The I2C-bus system clock (ϕIIC) is stopped, and the internal counter and flags are reset.
- The TOF bit in the I2CCR2 register becomes 0 (timeout not detected).

23.1.3.3 DFS Bit

The DFS bit enables the automatic recognition of a slave address. When the DFS bit is set to 0, the addressing format is selected and the slave address is automatically recognized. In this setting, data is received only when a general call address is received or a slave address match is detected. When the DFS bit is set to 1, the free data format is selected. In this setting, the slave address is not recognized, so all data are received.

23.1.3.4 RST Bit

The RST bit resets the I2C-bus interface when a communication error occurs. When the ICE bit is set to 1 (I2C-bus interface enabled), writing 1 (reset) to the RST bit has the following effects on the I2C-bus interface:

- Bits ADZ, AAS, AL, BBSY, TRS, and MST in the I2CSR register become 0, and the IRF bit becomes 1.
- The TOF bit in the I2CCR2 register becomes 0 (timeout not detected).
- The internal counter and flags are reset.

When the RST bit is written with 1, the multi-master I2C-bus interface is reset within a maximum of 2.5 ϕIIC cycles. Consequently, the RST bit automatically becomes 0.

Figure 23.6 shows the timing when the I2C-bus interface is reset.

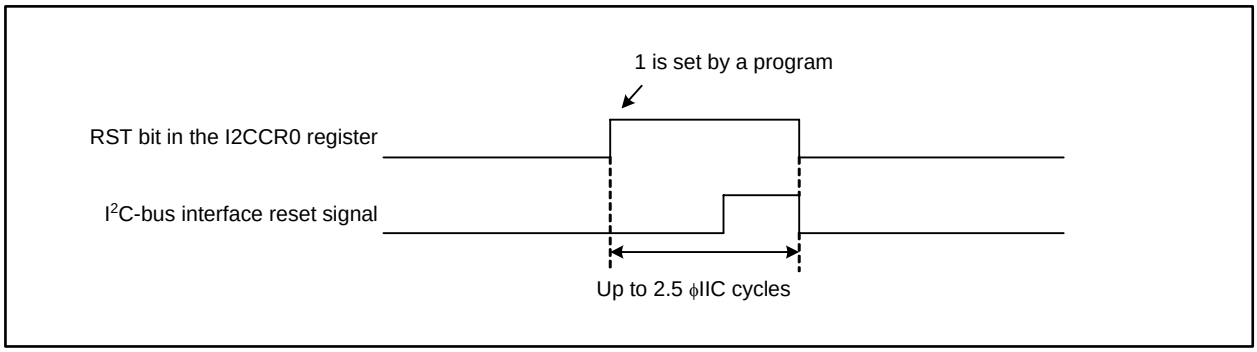


Figure 23.6 I2C-bus Interface Reset Timing

23.1.4 I2C-bus Clock Control Register (I2CCCR)

I2C-bus Clock Control Register

b7	b6	b5	b4	b3	b2	b1	b0

Symbol	Address	Reset Value
I2CCCR	044404h	0000 0000b

Bit Symbol	Bit Name	Function	RW
CKS0	Transmit/Receive Clock Frequency Control Bit	The transmit/receive clock frequency is given by ϕIIC/8n [Hz] in Standard-mode, or ϕIIC/4n [Hz] in Fast-mode, where n is a setting value. However, when 00101b is set in Fast-mode, the transmit/receive clock frequency becomes ϕIIC/2n [Hz]. Do not set to 00000b to 00010b	RW
CKS1			RW
CKS2			RW
CKS3			RW
CKS4			RW
CLKMD	Clock Mode Select Bit	0: Standard-mode 1: Fast-mode	RW
ACKD	ACK Data Bit	0: ACK sent 1: NACK sent	RW
ACKCLK	ACK Clock Generating Bit	0: ACK clock not generated 1: ACK clock generated	RW

Figure 23.7 I2CCCR Register

The I2CCCR register controls ACK and sets SCL mode and SCL clock frequency. While data is being transmitted or received, only rewrite the ACKD bit.

23.1.4.1 Bits CKS4 to CKS0

Bits CKS4 to CKS0 set the SCL clock frequency. The SCL clock frequency varies as shown in the Table 23.3, where n is a setting value of bits CKS4 to CKS0 (n = 3 to 31). Do not rewrite these bits while data is being transmitted or received.

Table 23.3 I2CCCR Register Setting Values and SCL Frequencies

Bits CKS4 to CKS0 Setting Value (n)	SCL Frequency (When ϕIIC = 4 MHz) (1)	
	Standard-mode	Fast-mode
0 to 2	Do not set (2)	Do not set (2)
3	Do not set (3)	333 kHz (ϕIIC/4n)
4	Do not set (3)	250 kHz (ϕIIC/4n)
5	100 kHz (ϕIIC/8n)	400 kHz (ϕIIC/2n) (4)
6 to 31	83 to 16 kHz (ϕIIC/8n)	166 to 32 kHz (ϕIIC/4n)

Notes:
1. The CKS value must be set so the SCL clock frequency is 100 kHz or less in Standard-mode or 400 kHz or less in Fast-mode. The high period of the SCL clock has a margin of error of +2 to -4 ϕIIC in Standard-mode, and +2 to -2 ϕIIC in Fast-mode. Note that if the high period is shortened, the low period is lengthened, so the frequency remains unchanged.
2. Do not set the CKS value to 0 to 2 regardless of the ϕIIC frequency.
3. When ϕIIC is 4 MHz or higher, do not set the CKS value to 3 or 4. The SCL clock frequency will extend beyond the specified range.
4. The normal duty cycle of the SCL clock is 50%. When the CKS value is 5 in Fast-mode, it varies from 35% to 45%.

23.1.4.2 CLKMD Bit

Set the CLKMD bit to select the SCL mode. Set this bit to 0 to select Standard-mode and 1 for Fast-mode. To use the device under the Fast-mode I2C-bus specification (up to 400 kbit/s), set ϕIIC to be 4 MHz or higher.

23.1.4.3 ACKD Bit

Set the ACKD bit to select the state of the MSDA pin with the ACK clock. When the ACKD bit is set to 0, the MSDA pin becomes low (acknowledged) by an ACK. When the ACKD bit is 1, the MSDA pin is held high with the ACK clock.
Table 23.4 lists the MSDA pin state with the ACK clock.

Table 23.4 MSDA Pin States with the ACK Clock

Received Content	DFS Bit	ACKD Bit	Slave Address	MSDA Pin State
Slave address	0	0	Match	Low (ACK)
			No match	High (NACK)
		1	—	High (NACK)
	1	0	—	Low (ACK)
		1	—	High (NACK)
Data	—	0	—	Low (ACK)
		1	—	High (NACK)

23.1.4.4 ACKCLK Bit

Set the ACKCLK bit to select whether or not to generate an ACK handshake. When this bit is 1 (ACK clock generated), an ACK clock pulse is generated after 1 byte of data is transmitted or received. When this bit is 0 (ACK clock not generated), the ACK clock is not generated after 1 byte of data is transmitted or received. In this case, the IR bit in the I2CIC register becomes 1 (I2C-bus interface interrupt requested) on the last falling edge of the clock for data transmission or reception.

23.1.5 I2C-bus START and STOP Conditions Control Register (I2CSSCR)

I2C-bus START and STOP Conditions Control Register

b7	b6	b5	b4	b3	b2	b1	b0

Symbol	Address	Reset Value
I2CSSCR	044405h	0001 1010b

Bit Symbol	Bit Name	Function	RW
SSC0	START and STOP Conditions Detection Setting Bit	The conditions for detecting START and STOP conditions (SCL open, set-up, and hold times) are set with these bits	RW
SSC1			RW
SSC2			RW
SSC3			RW
SSC4			RW
SIP	I2C-bus line Interrupt Pin Edge Select Bit	0: Falling edge 1: Rising edge	RW
SIS	I2C-bus line Interrupt Pin Select Bit	0: MSDA pin 1: MSCL pin	RW
STSPSEL	START and STOP Conditions Generating Mode Select Bit	0: Short mode 1: Long mode	RW

Figure 23.8 I2CSSCR Register

The I2CSSCR register controls the detection and generation of START and STOP conditions.

23.1.5.1 Bits SSC4 to SSC0

Bits SSC4 to SSC0 select the parameters for detecting the START and STOP conditions by setting the high period of SCL pin, set-up, and hold times. This parameter is set by referencing the I2C-bus system clock (ϕIIC). Therefore, it will change according to the XIN frequency and the setting of the I2C-bus system clock select bits (i.e. bits ICK4 to ICK0 in registers I2CCR2 and I2CCR1). Do not set an odd number or 00000b to bits SSC4 to SSC0. To detect a START or STOP condition, set the ICE bit in the I2CCR0 register to 1 (I2C-bus interface enabled). Table 23.11 lists the recommended values for bits SSC4 to SSC0.

23.1.5.2 SIP Bit

Set the SIP bit to select which of the edges of MSCL or MSDA pin generates the I2C-bus line interrupt. Set this bit to 0 to select the falling edge, and 1 to select the rising edge.

23.1.5.3 SIS Bit

Set the SIS bit to select the input signal to be used as an I2C-bus line interrupt source. To select the MSDA pin as an I2C-bus line interrupt source, set this bit to 0. To select the MSCL pin, set this bit to 1.

23.1.5.4 STSPSEL Bit

Set the STSPSEL bit to select the set-up and hold times when START and STOP conditions are generated. Set this bit to 0 to select short mode and 1 to select long mode. The STSPSEL bit must be set to 1 (long mode) when the ϕIIC frequency is higher than 4 MHz. Figure 23.16 shows the START condition generation timing. Table 23.9 lists the set-up and hold times when START and STOP conditions are generated.

23.1.6 I2C-bus Control Register 1 (I2CCR1)

I2C-bus Control Register 1 (1)

b7	b6	b5	b4	b3	b2	b1	b0
				0	0		

Symbol: I2CCR1 | Address: 044406h | Reset Value: 0011 0000b

Bit Symbol	Bit Name	Function	RW
STIE	STOP Condition Detection Interrupt Enable Bit	0: Disabled 1: Enabled	RW
RIE	Successful Receive Interrupt Enable Bit (2)	0: Disabled 1: Enabled	RW
— (b3-b2)	Reserved	Should be written with 0	RW
SDAO	Internal SDA Output Monitor Bit	0: Low 1: High	RO
SCLO	Internal SCL Output Monitor Bit	0: Low 1: High	RO
ICK0	I2C-bus System Clock Select Bit (3)	b7 b6 0 0 : fIIC divided-by-2 0 1 : fIIC divided-by-4 1 0 : fIIC divided-by-8 1 1 : Do not use this combination	RW
ICK1			RW

Notes:
1. Do not use a bit processing instruction with this register.
2. Set this bit to 0 when the ACKCLK is 0 (ACK clock is not generated).
3. These bits are enabled when bits ICK4 to ICK2 in the I2CCR2 register are 000b.

Figure 23.9 I2CCR1 Register

The I2CCR1 register controls the I2C-bus interface.

23.1.6.1 STIE Bit

The STIE bit enables the STOP condition detection interrupt. Set this bit to 1 to enable the I2C-bus interface interrupt when a STOP condition is detected. Consequently, the STOP bit in the I2CCR2 register becomes 1 (STOP condition detection interrupt requested) and the IR bit in the I2CIC register becomes 1 (I2C-bus interface interrupt requested).

23.1.6.2 RIE Bit

When the ACKCLK bit in the I2CCCR register is 1 (ACK clock generated), the RIE bit enables the interrupt which is generated when receiving the last bit of data. When the RIE bit is 1, the I2C-bus interface interrupt is generated when the last bit (the eighth falling edge of the SCL) of data is received.

Note that the I2C-bus interface interrupt is always generated when the ACK bit (the ninth falling edge of the SCL) is received regardless of the RIE bit setting. Therefore, when the RIE bit is set to 1, two I2C-bus interface interrupts are generated per data. The source of the interrupt can be identified by reading the RIE bit. The read value indicates the internal WAIT flag state. When the read value is 1, the last bit of data is the interrupt source. When the read value is 0, the ACK bit is the interrupt source.

Set the RIE bit to 0 when the ACKCLK bit in the I2CCCR register is 0 (ACK clock not generated). When the device is transmitting data or receiving a slave address, the I2C-bus interface interrupt is generated only by the ACK bit (the ninth falling edge of the SCL) regardless of the RIE bit setting. In both cases, the internal WAIT flag is 0.

Table 23.5 I2C-bus Interrupt Request Generating Timings when Data are Received and Resuming Communication

I2C-bus Interface Interrupt Generating Timing	Internal WAIT Flag	Resuming Transmission/Reception
Last bit of data (on eighth clock)	1	Write to the ACKD bit in the I2CCCR register
ACK bit (on ninth clock)	0	Write to the I2CTRSR register

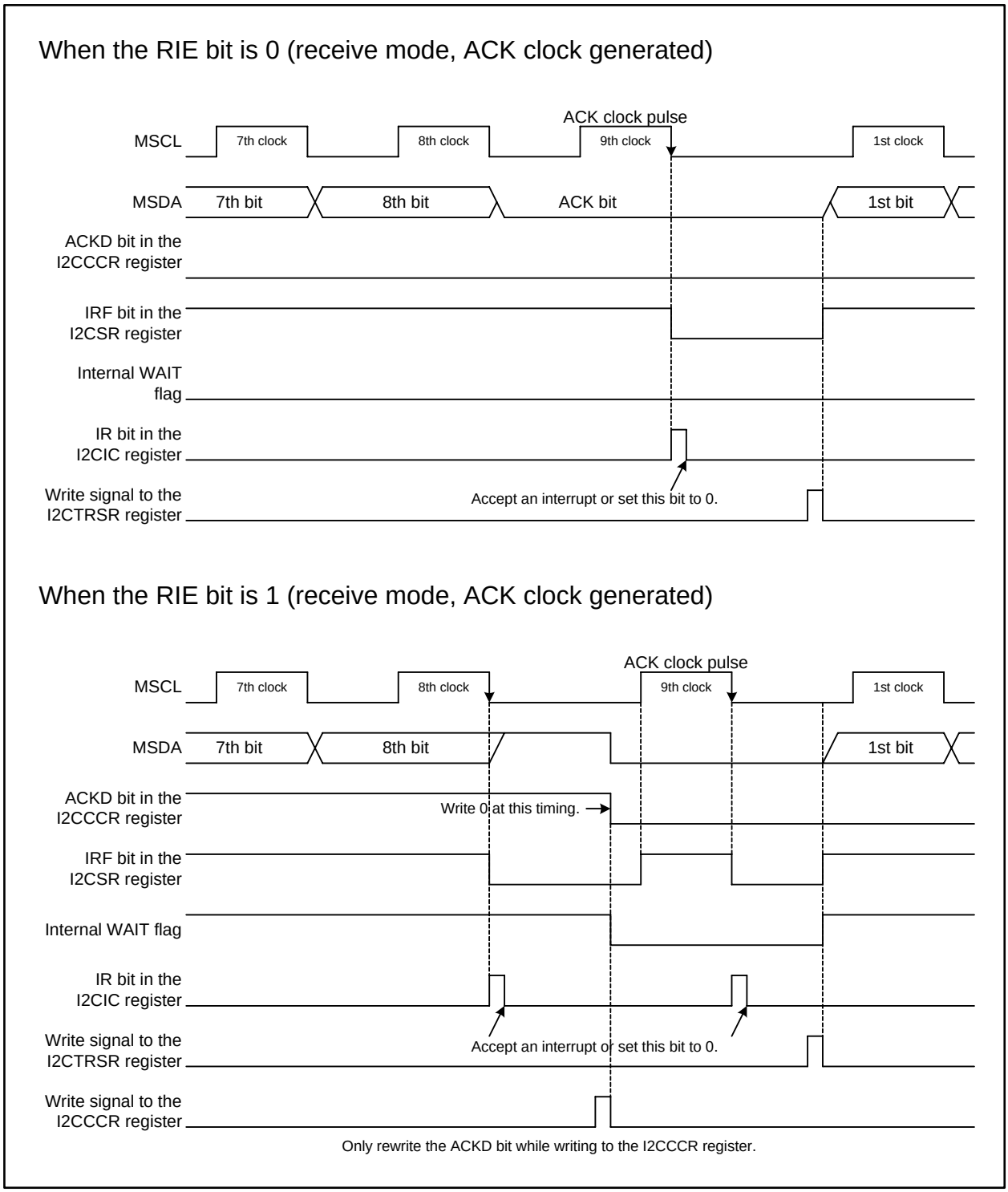

When the RIE bit is 0 (receive mode, ACK clock generated)
ACK clock pulse
MSCL
7th clock
8th clock
9th clock
1st clock
MSDA
7th bit
8th bit
ACK bit
1st bit
ACKD bit in the I2CCCR register
IRF bit in the I2CSR register
Internal WAIT flag
IR bit in the I2CIC register
Write signal to the I2CTRSR register
Accept an interrupt or set this bit to 0.
When the RIE bit is 1 (receive mode, ACK clock generated)
ACK clock pulse
MSCL
7th clock
8th clock
9th clock
1st clock
MSDA
7th bit
8th bit
1st bit
ACKD bit in the I2CCCR register
Write 0 at this timing.
IRF bit in the I2CSR register
Internal WAIT flag
IR bit in the I2CIC register
Write signal to the I2CTRSR register
Accept an interrupt or set this bit to 0.
Write signal to the I2CCCR register
Only rewrite the ACKD bit while writing to the I2CCCR register.

Figure 23.10 Interrupt Request Generating Timing in Receive Mode

23.1.6.3 Bits SDAO and SCLO

Bits SDAO and SCLO are read-only bits, and are used to monitor the logical values of the internal SDA output signal and internal SCL output signal, respectively. Only set these bits to 0. Note that the levels of the internal SDA and SCL output signals read from bits SDAO and SCLO are pre-influenced by the external devices, and do not indicate MSDA and MSCL pin states.

23.1.6.4 Bits ICK1 and ICK0

Set bits ICK1 and ICK0 to select the I2C-bus system clock frequency (ϕIIC). These bits are enabled when bits ICK4 to ICK2 in the I2CCR2 register are 000b. Rewrite these bits when the ICE bit in the I2CCR0 register is 0 (I2C-bus interface disabled). The I2C-bus system clock frequency (ϕIIC) is selected from fIIC divided-by-2, -4, and -8 by setting these bits. fIIC divided-by-2.5, -3, -5, and -6 are also available by setting bits ICK4 to ICK2 in the I2CCR2 register. However, bits ICK1 and ICK0 are disabled in this case.

Table 23.6 I2C-bus System Clock (ϕIIC) Select Bit Settings

I2CCR2 Register			I2CCR1 Register		ϕIIC
ICK4 bit	ICK3 bit	ICK2 bit	ICK1 bit	ICK0 bit	
0	0	0	0	0	fIIC divided-by-2
			0	1	fIIC divided-by-4
			1	0	fIIC divided-by-8
0	0	1	0	0	fIIC divided-by-2.5
0	1	0	0	0	fIIC divided-by-3
0	1	1	0	0	fIIC divided-by-5
1	0	0	0	0	fIIC divided-by-6

Only set the values listed above.

23.1.7 I2C-bus Control Register 2 (I2CCR2)

I2C-bus Control Register 2

b7	b6	b5	b4	b3	b2	b1	b0
	0						

Symbol: I2CCR2 | Address: 044407h | Reset Value: 0X00 0000b

Bit Symbol	Bit Name	Function	RW
TOE	Timeout Detector Enable Bit	0: Timeout detector disabled 1: Timeout detector enabled	RW
TOF	Timeout Detect Flag	0: Timeout not detected 1: Timeout detected	RO
TOSEL	Timeout Detect Period Select Bit	0: Long 1: Short	RW
ICK2	I2C-bus System Clock Select Bit	b5 b4 b3 0 0 0 : ϕIIC = set by bits ICK1 and ICK0 in the I2CCR1 register 0 0 1 : ϕIIC = fIIC divided-by-2.5 0 1 0 : ϕIIC = fIIC divided-by-3 0 1 1 : ϕIIC = fIIC divided-by-5 1 0 0 : ϕIIC = fIIC divided-by-6 Only set the values listed above	RW
ICK3			RW
ICK4			RW
— (b6)	Reserved	Should be written with 0	RW
STOP	STOP Condition Detect Interrupt Request Monitor Bit	0: I2C-bus interface interrupt not requested 1: I2C-bus interface interrupt requested	RW

Figure 23.11 I2CCR2 Register

The I2CCR2 register controls communication error detection. If the SCL clock stops, each device connected to the bus is halted suspending communication. To avoid this, the multi-master I2C-bus interface supports a function to generate an I2C-bus interface interrupt when the SCL clock is held high for a specified period of time during transmission or reception.

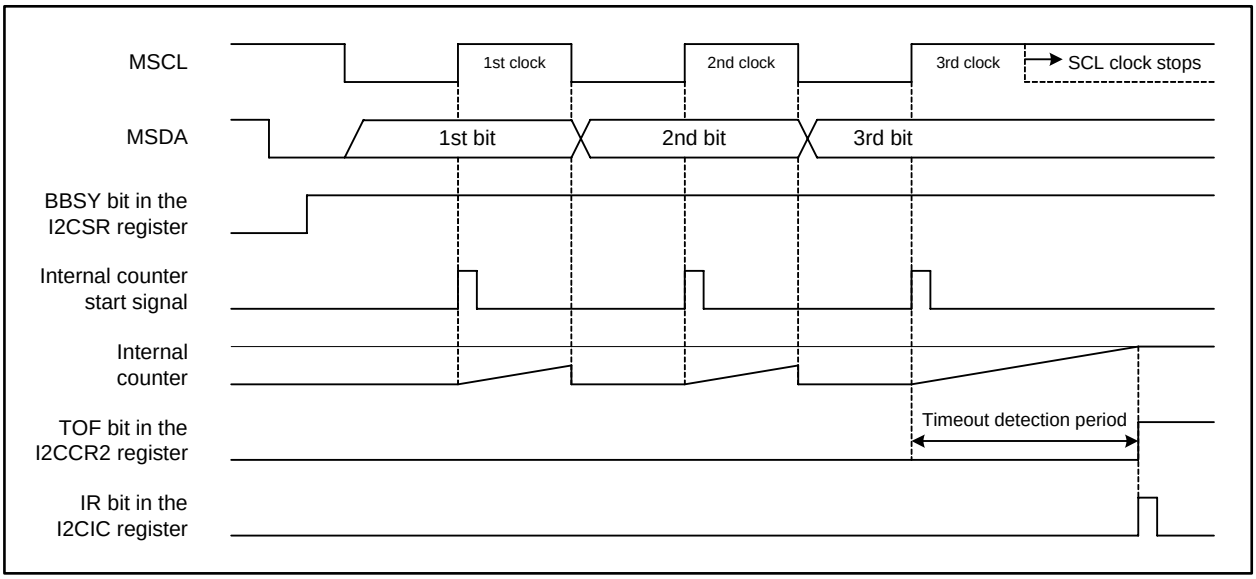


Figure 23.12 Timeout Detecting Timing

23.1.7.1 TOE Bit

The TOE bit enables the timeout detector. When this bit is set to 1, the timeout detector is enabled, and when the SCL clock is held high for a specified period of time while the BBSY bit in the I2CSR register is 1 (bus is busy), an I2C-bus interface interrupt request is generated.

The timeout detection period is determined by 1) the internal counter that uses ϕIIC as a count source, and 2) the TOSEL bit setting (selects the timeout detection period to be either long or short). Refer to 23.1.7.3 “TOSEL bit” for details.

When a timeout is detected, set the ICE bit in the I2CCR0 register to 0 (I2C-bus interface disabled) and initialize the I2C-bus interface.

23.1.7.2 TOF Bit

The TOF bit is a flag that indicates the state of a timeout detection. This bit is enabled when the TOE bit is 1. When the TOF bit becomes 1 (timeout detected), the IR bit in the I2CIC register becomes 1 (I2C-bus interface interrupt requested) simultaneously.

23.1.7.3 TOSEL Bit

The TOSEL bit selects a long or short length for a timeout detection period. This bit is enabled when the TOE bit is 1 (timeout detector enabled). Set this bit to 0 to select the long timeout period. In this setting, the internal counter functions as a 16-bit counter. Set this bit to 1 to select the short timeout period. In this setting, the internal counter functions as a 14-bit counter.

The internal counter increments using the I2C-bus system clock (ϕIIC) as a count source.

Table 23.7 lists timeout detection periods.

Table 23.7 Example Timeout Detection Periods

ϕIIC	Long Timeout Detection Period (TOSEL = 0)	Short Timeout Detection Period (TOSEL = 1)
4 MHz	16.4 ms	4.1 ms
2 MHz	32.8 ms	8.2 ms
1 MHz	65.6 ms	16.4 ms

23.1.7.4 Bits ICK4 to ICK2

Set bits ICK4 to ICK2 to select the I2C-bus system clock frequency (ϕIIC). Rewrite these bits when the ICE bit in the I2CCR0 register is 0 (I2C-bus interface disabled).

The I2C-bus system clock frequency (ϕIIC) can be selected from fIIC divided-by-2.5, -3, -5, and -6. Or, when bits ICK4 to ICK2 are 000b, the I2C-bus system clock frequency (ϕIIC) can be selected from fIIC divided-by-2, -4, and -8 by setting bits ICK1 and ICK0 in the I2CCR1 register. Refer to Table 23.6.

23.1.7.5 STOP Bit

The STOP bit monitors the STOP condition detection interrupt. When the I2C-bus interface interrupt is generated by the detection of a STOP condition, the STOP bit becomes 1. This bit is enabled when the STIE bit in the I2CCR1 register is 1 (STOP condition detection interrupt is enabled). This bit can only be set to 0. Writing 1 to this bit has no effect.

23.1.8 I2C-bus Status Register (I2CSR)

I2C-bus Status Register

b7	b6	b5	b4	b3	b2	b1	b0

Symbol: I2CSR
Address: 044408h
Reset Value: 0001 000Xb

Bit Symbol	Bit Name	Function	RW
LRB	Last Received Bit (1, 2)	0: Last received bit is 0 1: Last received bit is 1	RW
ADZ	General Call Address Detect Flag (1, 2)	0: General call address not detected 1: General call address detected	RW
AAS	Slave Address Match Flag (1, 2)	0: Address not matched 1: Address matched	RW
AL	Arbitration Lost Detect Flag (1, 2)	0: Arbitration lost not detected 1: Arbitration lost detected	RW
IRF	I2C-bus Interface Interrupt Request Flag (3)	0: Requested 1: Not requested	RO
BBSY	Bus Busy Flag (2)	0: Bus is free 1: Bus is busy	RW
TRS	Transmit/Receive Switch Bit	0: Receive mode 1: Transmit mode (1)	RW
MST	Master/Slave Select Bit	0: Slave mode 1: Master mode (1)	RW

Notes:
1. Write 1111b to the lower four bits of this register to set the TRS or MST bit to 1 without generating a START or STOP condition.
2. These bits are read-only when using them to check the status.
3. This bit is read-only. Only set this bit to 0.

Figure 23.13 I2CSR Register

The I2CSR register monitors the state of the I2C-bus interface. Write to this register only when using the functions listed in Table 23.8, and only set the values that are listed. Note that the lower six bits are not rewritten even though a value from Table 23.8 is written.

Table 23.8 I2CSR Register Settings and Functions

Values Written to the I2CSR Register								Function
MST	TRS	BBSY	IRF	AL	AAS	ADZ	LRB	
0	0	X	0	1	1	1	1	Select slave-receive mode
0	1							Select slave-transmit mode
1	0							Select master-receive mode
1	1							Select master-transmit mode
1	1	0	0	0	0	0	0	Select master-transmit mode and set the device to be on STOP condition standby.
		1						Select master-transmit mode and set the device to be on START condition standby.

23.1.8.1 LRB Bit

The LRB bit stores the data of the last received bit. It is used to check whether an ACK is received. When the ACKCLK bit in the I2CCCR register is 1 (ACK clock generated), the LRB bit becomes 0 when the ACK is received, and 1 when the ACK is not received. When the ACKCLK bit is 0 (ACK clock not generated), the last bit of data is stored to the LRB bit. When a value is written to the I2CTRSR register, the LRB bit becomes 0.

23.1.8.2 ADZ Bit

The ADZ bit is a flag that indicates that the general call address was received. When the DFS bit in the I2CCR0 register is 0 (addressing format) in slave-receive mode, the ADZ bit becomes 1 when the general call address is received.
The ADZ bit becomes 0 in any of the following cases:

- When a STOP or START condition is detected
- When the ICE bit in the I2CCR0 register is set to 0 (I2C-bus interface disabled)
- When the RST bit in the I2CCR0 register is written with 1 (I2C-bus interface reset)

23.1.8.3 AAS Bit

The AAS bit is a flag that indicates whether the received address matches its own slave address. The AAS bit becomes 1 when the received address matches its own slave address in bits SAD6 to SAD0 in the I2CSAR register, when the DFS bit in the I2CCR0 register is 0 (addressing format) in slave-receive mode, or when the received address is the general call address.
The AAS bit becomes 0 in any of the following cases:

- When data is written to the I2CTRSR register
- When the ICE bit in the I2CCR0 register is set to 0 (I2C-bus interface disabled)
- When the RST bit in the I2CCR0 register is written with 1 (I2C-bus interface reset)

23.1.8.4 AL Bit

The AL bit is a flag that indicates arbitration lost detection. In master transmit mode, if the MSDA pin is changed to low by another device, then the AL bit becomes 1. Consequently, the TRS bit in the I2CSR register becomes 0 (receive mode), and then the MST bit becomes 0 (slave mode) at the end of the byte in which an arbitration lost is detected.
The AL bit becomes 0 in any of the following cases:

- When data is written to the I2CTRSR register
- When the ICE bit in the I2CCR0 register is set to 0 (I2C-bus interface disabled)
- When the RST bit in the I2CCR0 register is written with 1 (I2C-bus interface reset)

23.1.8.5 IRF Bit

Set the IRF bit to generate the I2C-bus interface interrupt request signal. When the I2C-bus interface interrupt source is generated, first the IRF bit becomes 0, then the I2C-bus interface interrupt is generated on the falling edge of the IRF bit. Refer to Figure 23.10 for the timing.

The IRF bit becomes 0 in any of the following cases:

- When 1-byte data transmission is completed (including when an arbitration lost is detected)
- When 1-byte data reception is completed
- When the slave address is matched in addressing format in slave-receive mode
- When the general call address is received in addressing format in slave-receive mode
- When address data reception is completed in free data format in slave-receive mode

The IRF bit becomes 1 in any of the following cases:

- When data is written to the I2CTRSR register
- When data is written to the I2CCCR register (internal WAIT flag is 1)
- When the ICE bit in the I2CCR0 register is set to 0 (I2C-bus interface disabled)
- When the RST bit in the I2CCR0 register is written with 1 (I2C-bus interface reset)

23.1.8.6 BBSY Bit

The BBSY bit is a flag that indicates the availability of the I2C-bus. The BBSY bit becomes 1 when a START condition is detected, and 0 when a STOP condition is detected. When the BBSY bit is 0, the I2C-bus is not in use, and is available for the device to generate a START condition.

The detection of a START or STOP condition is dependent on the setting of bits SSC4 to SSC0 in the I2CSSCR register.

The BBSY bit becomes 0 in any of the following cases:

- When a STOP condition is detected
- When the ICE bit in the I2CCR0 register is set to 0 (I2C-bus interface disabled)
- When the RST bit in the I2CCR0 register is written with 1 (I2C-bus interface reset)

23.1.8.7 TRS Bit

The TRS bit determines the direction of data communication. When this bit is set to 0, the device enters receive mode and waits for data to be sent from another device. When this bit is set to 1, the device enters transmit mode and transmits data and address to the SDA line synchronized with the SCL clock.

The TRS bit automatically becomes 1 (transmit mode) when the received address matches its own slave address and the received R/$\overline{W}$ bit is 1 (data requested) in addressing format in slave-receive mode.

The TRS bit becomes 0 in any of the following cases:

- When this bit is set to 0
- When an arbitration lost is detected
- When a STOP condition is detected
- When the START condition redundancy prevention function is activated
- When a START condition is detected in slave mode
- When a NACK is received in slave mode
- When the ICE bit in the I2CCR0 register is set to 0 (I2C-bus interface disabled)
- When the RST bit in the I2CCR0 register is written with 1 (I2C-bus interface reset)

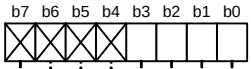
23.1.8.8 MST Bit

Set the MST bit to select master or slave mode. To enter slave mode, set this bit to 0. Communication is initiated in synchronization with the SCL clock generated by the master device. Set this bit to 1 to enter master mode. The device generates the SCL clock to initiate communication.
The MST bit becomes 0 in any of the following cases:

- When the MST bit is set to 0
- When an arbitration lost is detected, and transmission of the corresponding byte is completed
- When a STOP condition is detected
- When a START condition is detected
- When the START condition redundancy prevention function is enabled
- When the ICE bit in the I2CCR0 register is set to 0 (I2C-bus interface disabled)
- When the RST bit in the I2CCR0 register is written with 1 (I2C-bus interface reset)

23.1.9 I2C-bus Mode Register (I2CMR)

I2C-bus Mode Register (1)



Symbol	Address	Reset Value
I2CMR	044410h	XXXX 0000b

Bit Symbol	Bit Name	Function	RW
I2CEN	I2C-bus Interface/UART2 Switch Bit	0: UART2 1: I2C-bus interface	RW
CLK0	I2C-bus Interface Clock Source Select Bit	b3 b2 b1 0 0 0 : f1 divided-by-2 0 0 1 : f8 divided-by-2 0 1 0 : f2n divided-by-2 0 1 1 : Do not use this combination 1 0 0 : f1 1 0 1 : f8 1 1 0 : f2n 1 1 1 : Do not use this combination	RW
CLK1			RW
CLK2			RW
— (b7-b4)	No register bits; should be written with 0 and read as undefined value		RW

Note:
1. Set the PRC1 bit in the PRCR register to 1 (write enabled) before rewriting this register.

Figure 23.14 I2CMR Register

The I2CMR register selects signals for the I2C-bus interface and to select the clock source. Set the PRC1 bit in the PRCR register to 1 (write enabled) before rewriting this register.

23.1.9.1 I2CEN Bit

The I2CEN bit switches between signals for UART2 and the I2C-bus interface. Set this bit to 1 to use the following signals: MSDA, MSCL, the I2C-bus interface interrupt, and the I2C-bus line interrupt. When this bit is set to 0, signals for UART2 are enabled.

23.1.9.2 Bits CLK2 to CLK0

Bits CLK2 to CLK0 select the clock source for the I2C-bus interface clock (fIIC). It is selected from f1 divided-by-2, f8 divided-by-2, f2n divided-by-2, f1, f8, or f2n.

The clock source selected for the I2C-bus interface (fIIC) is used as the clock source for the I2C-bus system clock (ϕIIC).

23.2 Generating a START Condition

To enter a START condition standby state, write E0h to the I2CSR register while the ICE bit in the I2CCR0 register is 1 (I2C-bus interface enabled) and the BBSY bit in the I2CSR register is 0 (bus is free). When in standby, write a slave address to the I2CTRSR register to generate a START condition. Consequently, the bit counter becomes 000b, 1 byte of the SCL clock is output, and the slave address is transmitted. Figure 23.15 shows how to generate a START condition.

Note that after a STOP condition is generated, writing to the I2CSR register is disabled for 1.5 cycles of ϕIIC after the BBSY bit becomes 0 (bus is free). To generate a START condition immediately after generating a STOP condition, first write E0h to the I2CSR register, then confirm that bits STR and MST in the I2CSR register are 1. After that, write a slave address to the I2CTRSR register.

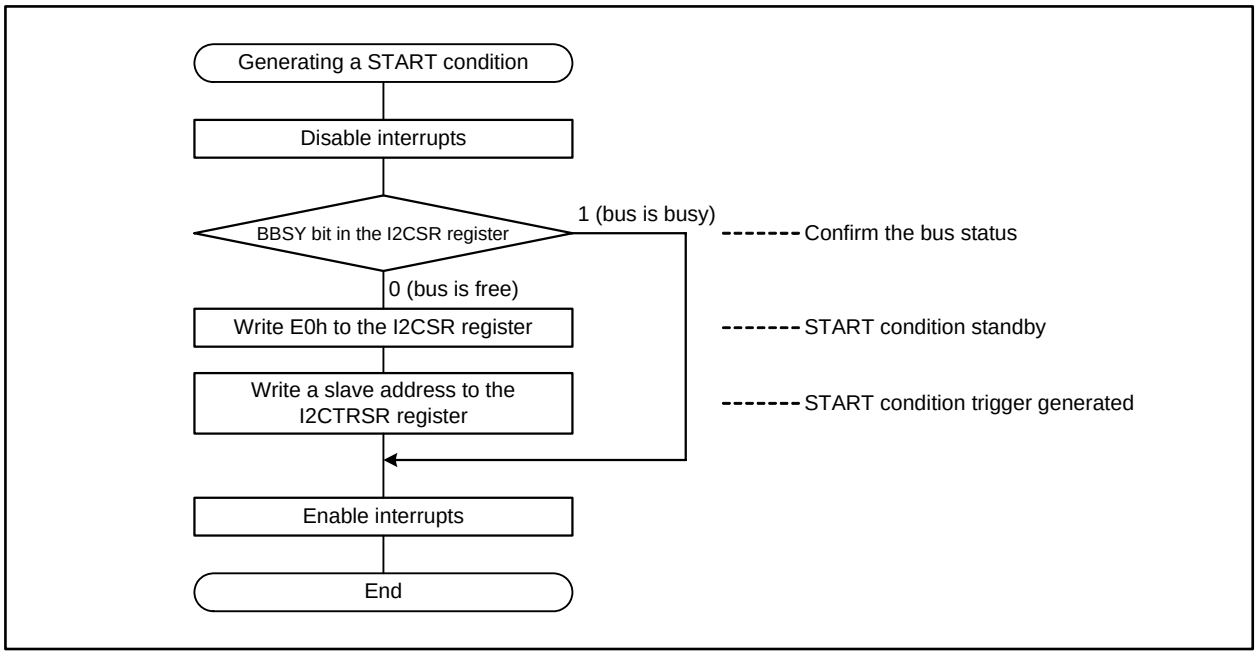

Generating a START condition
Disable interrupts
BBSY bit in the I2CSR register
1 (bus is busy)
Confirm the bus status
0 (bus is free)
Write E0h to the I2CSR register
START condition standby
Write a slave address to the I2CTRSR register
START condition trigger generated
Enable interrupts
End

Figure 23.15 Generating a START Condition

The timing to generate a START condition differs between Standard-mode and Fast-mode. Figure 23.16 shows START condition generating timing. Table 23.9 lists the set-up and hold times when a START or STOP condition is generated.

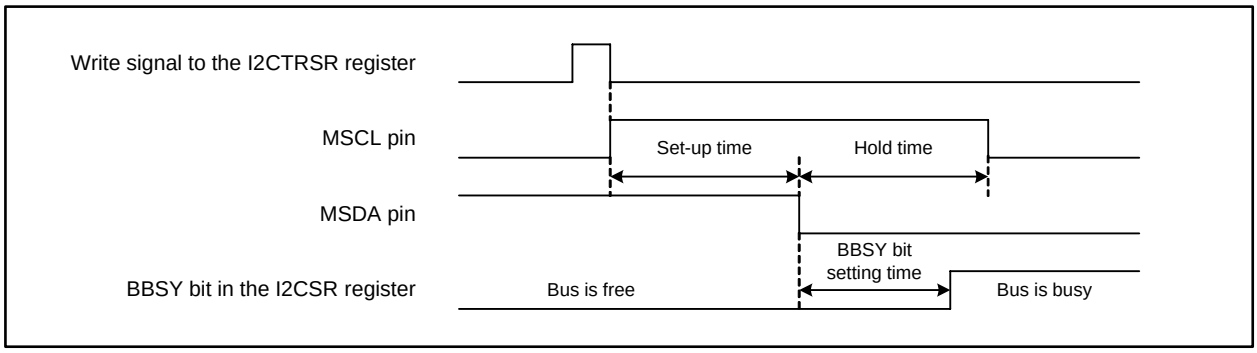


Figure 23.16 START Condition Generating Timing

Table 23.9 Set-up and Hold Times When Generating a START or STOP Condition

Parameter	SCL Mode	Short Mode (STSPSEL = 0)	Long Mode (STSPSEL = 1)
Set-up time	Standard-mode (CLKMD = 0)	5.0 µs (20)	13.0 µs (52)
	Fast-mode (CLKMD = 1)	2.5 µs (10)	6.5 µs (26)
Hold time	Standard-mode (CLKMD = 0)	5.0 µs (20)	13.0 µs (52)
	Fast-mode (CLKMD = 1)	2.5 µs (10)	6.5 µs (26)

CLKMD: Bit in the I2CCCR register
STSPSEL: Bit in the I2CSSCR register
ϕIIC cycle numbers are in parentheses.

23.3 Generating a STOP Condition

To enter a STOP condition standby state, write C0h to the I2CSR register while the ICE bit in the I2CCR0 register is 1 (I2C-bus interface enabled). Consequently, the MSDA pin becomes low. When in a standby state, write dummy data to the I2CTRSR register to generate a STOP condition. Figure 23.17 shows how to generate a STOP condition.

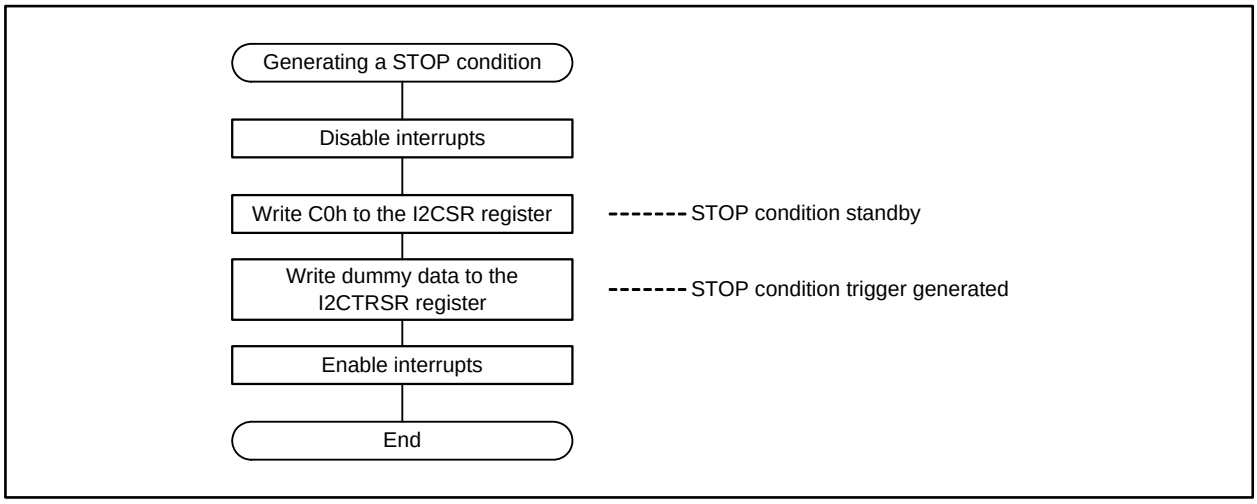


Figure 23.17 Generating a STOP Condition

The timing for generating a STOP condition differs between Standard-mode and Fast-mode. Figure 23.18 shows STOP condition generating timing. Table 23.9 lists the set-up and hold times when a START or STOP condition is generated.

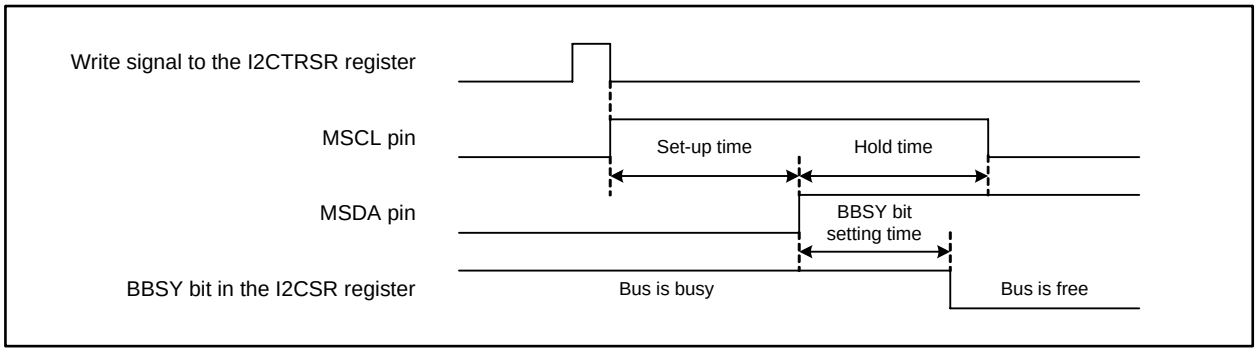


Figure 23.18 STOP Condition Generating Timing

To ensure the successful generation of a STOP condition, after the standby setting, do not write to the I2CSR or I2CTRSR register before the BBSY bit in the I2CSR register becomes 0 (bus is free), otherwise the STOP condition might not be generated successfully.
Furthermore, after the standby setting, if the MSCL pin input signal becomes low after the MSCL pin level becomes high, before the BBSY bit in the I2CSR register becomes 0 (bus is free), then the internal SCL output becomes low. In this case, low output from the MSCL pin is stopped (clock line released) by generating a STOP condition, by setting the ICE bit in the I2CCR0 register to 0 (I2C-bus interface disabled), or by setting the RST bit to 1 (I2C-bus interface reset)

23.4 START Condition Redundancy Prevention Function

A START condition is generated when the bus is free (confirmed with the BBSY bit in the I2CSR register). However, before a START condition is generated, if a different master device generates another START condition, the BBSY bit may become 1. In this case, the START condition redundancy prevention function terminates the generation of its own START condition.

The START condition redundancy prevention functions as follows:

- The START condition standby setting is disabled (exits from standby state)
- Writing to the I2CTRSR register is disabled (generation of the START condition trigger is disabled)
- Bits MST and TRS in the I2CSR register become 0 (enters into slave-receive mode)
- The AL bit in the I2CSR register becomes 1 (arbitration lost is detected)

Figure 23.19 shows the operation of the START condition redundancy prevention function.

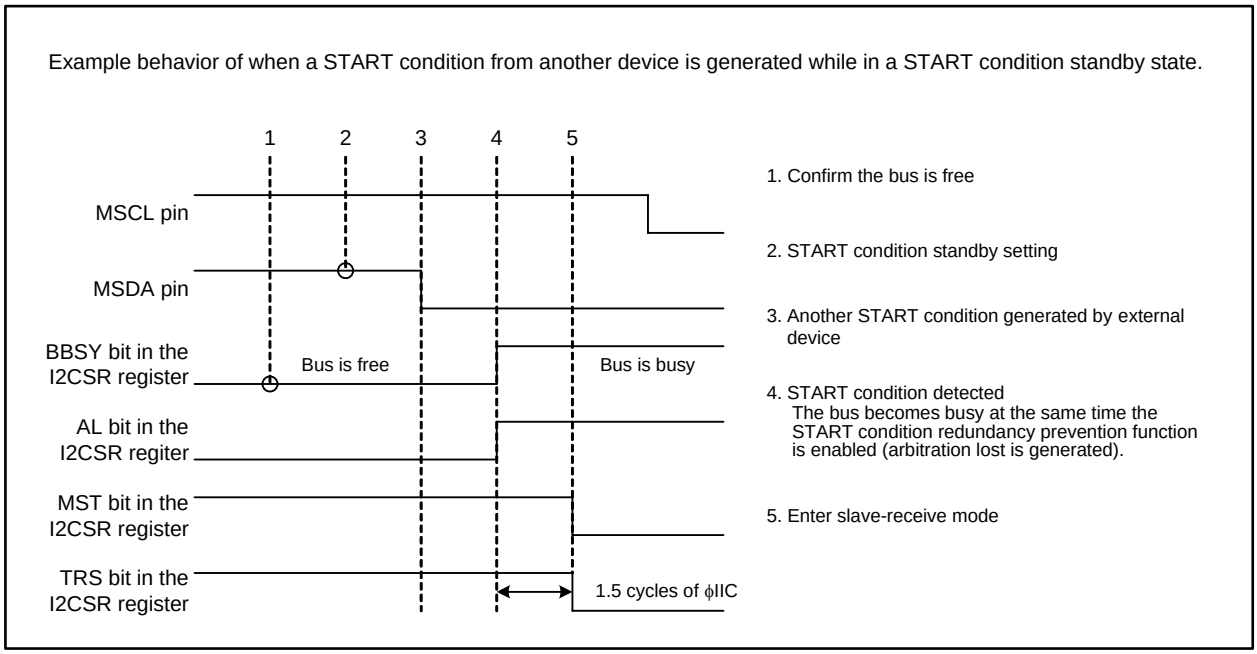

Example behavior of when a START condition from another device is generated while in a START condition standby state.
1
2
3
4
5
MSCL pin
MSDA pin
BBSY bit in the I2CSR register
AL bit in the I2CSR regiter
MST bit in the I2CSR register
TRS bit in the I2CSR register
Bus is free
Bus is busy
1.5 cycles of ϕIIC
1. Confirm the bus is free
2. START condition standby setting
3. Another START condition generated by external device
4. START condition detected
The bus becomes busy at the same time the START condition redundancy prevention function is enabled (arbitration lost is generated).
5. Enter slave-receive mode

Figure 23.19 Example Operation of the START Condition Redundancy Prevention Function

The START condition redundancy prevention function is enabled from the falling edge of an SDA line in a START condition until the slave address is completely received. This means, when registers I2CSR and I2CTRSR are written during this period, then the START condition redundancy prevention function is enabled. Figure 23.20 shows the duration.

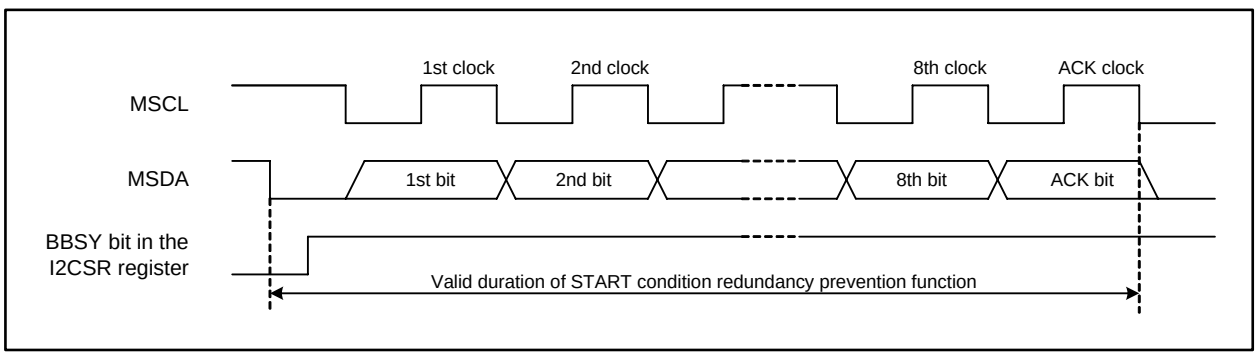


Figure 23.20 Enabled Duration of the START Condition Redundancy Prevention Function

23.5 Detecting START and STOP Conditions

Figure 23.21 shows START condition detection, Figure 23.22 shows STOP condition detection, and Table 23.10 lists the parameters for detecting START and STOP conditions. The parameters to detect START and STOP conditions are set with bits SSC4 to SSC0 in the I2CSSCR register. These parameters are detectable only when the input signals of pins MSCL and MSDA meet all the conditions of the high period of MSCL pin, set-up, and hold times in Table 23.10.

The BBSY bit in the I2CSR register becomes 1 when a START condition is detected, and 0 when a STOP condition is detected. The timing for setting the BBSY bit differs between Standard-mode and Fast-mode. Refer to Table 23.11 for BBSY bit setting time. Table 23.11 lists the recommended settings for bits SSC4 to SSC0 in Standard-mode.

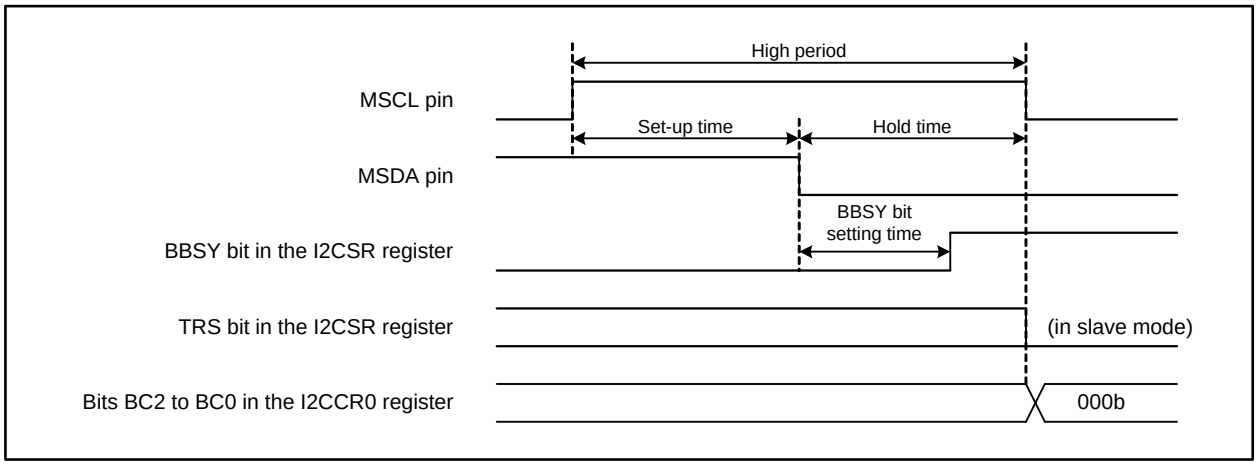


Figure 23.21 Detecting a START Condition

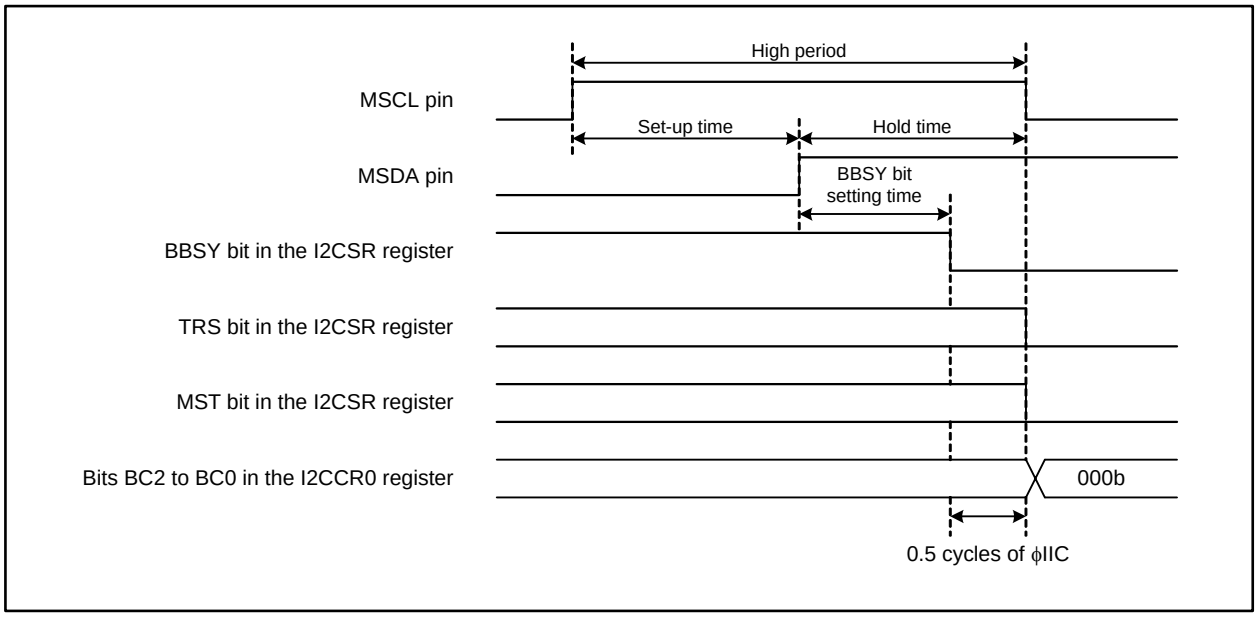


Figure 23.22 Detecting a STOP Condition

Table 23.10 Parameters for Detecting START and STOP Conditions

Parameter	Standard-mode	Fast-mode
High period of MSCL pin	$\frac{\text{SSC value}}{2}$ + 1 cycle (6.25 µs)	4 cycles (1.0 µs)
Set-up time	$\frac{\text{SSC value}}{2}$ + 1 cycle < 4.0 µs (3.25 µs)	2 cycles (0.5 µs)
Hold time	$\frac{\text{SSC value}}{2}$ cycles < 4.0 µs (3.0 µs)	2 cycles (0.5 µs)
BBSY bit set/reset time	$\frac{\text{SSC value - 1}}{2}$ + 2 cycles (3.375 µs)	3.5 cycles (0.875 µs)

Unit: ϕIIC cycles
SSC value: Setting value of bits SSC4 to SSC0 in the I2CSSCR register. Do not set these bits to 0 or an odd number.
Example times of when ϕIIC = 4 MHz and the I2CSSCR register = 18h are in parentheses.

Table 23.11 Recommended Values for Bits SSC4 to SSC0 in Standard-mode

ϕIIC	SSC Recommended Value	Parameters for Detecting START and STOP Conditions			BBSY Bit Set/Reset Time
		High period of MSCL pin	Set-up time	Hold time	
5 MHz	30	6.2 µs (31)	3.2 µs (16)	3.0 µs (15)	4.125 µs (16.5)
4 MHz	26	6.75 µs (27)	3.5 µs (14)	3.25 µs (13)	3.625 µs (14.5)
	24	6.25 µs (25)	3.25 µs (13)	3.0 µs (12)	3.375 µs (13.5)
2 MHz	12	6.5 µs (13)	3.5 µs (7)	3.0 µs (6)	3.75 µs (7.5)
	10	5.5 µs (11)	3.0 µs (6)	2.5 µs (5)	3.25 µs (6.5)
1 MHz	4	5.0 µs (5)	3.0 µs (3)	2.0 µs (2)	3.5 µs (3.5)

The number of ϕIIC cycles are in parentheses.
SSC recommended values: Decimal value of bits SSC4 to SSC0 in the I2CSSCR register.

23.6 Data Transmission and Reception

Examples of the data transmission and reception format for master-transmission or slave-reception in a 7-bit address format are shown in section 23.6.1 “Master Transmission” and 23.6.2 “Slave Reception”. These examples assume communication starts after initialization using the parameters set in Table 23.12.

Table 23.12 Example of Initial Settings

Register	Setting Value	Parameter	Initial Setting
I2CSAR	02h	Slave address	1
I2CCCR	85h	SCL frequency	100 kHz (ϕIIC = 4 MHz)
		Clock mode	Standard-mode
		ACK clock generation	ACK clock generated
I2CCR2	00h	Timeout Detector	Disabled
I2CCR1	13h	STOP condition detection interrupt	Enabled
		Successful data receive interrupt	Enabled
		ϕIIC	fIIC divided-by-2
I2CSR	0Fh	Communication mode	Slave-receive mode
I2CSSCR	98h	SSC value (see Table 23.11)	24
		START and STOP conditions generation mode	Long mode
I2CCR0	08h	Number of bits to be transmitted or received	8 bits
		I2C-bus interface	Enabled (communication enabled)
		Data format	Addressing format
I2CMR	09h	I2C-bus interface/UART2	I2C-bus interface selected
		I2C-bus interface clock source	fIIC = f2n

23.6.1 Master Transmission

The operation and procedures of master transmission are described in this section. Figure 23.23 shows an example of master transmission operation. For (A) to (C) in the figure, see A to C in the descriptions and procedures below. (1) to (3) show the program's instructions. Arrows indicate that the procedure is performed by the MCU automatically.

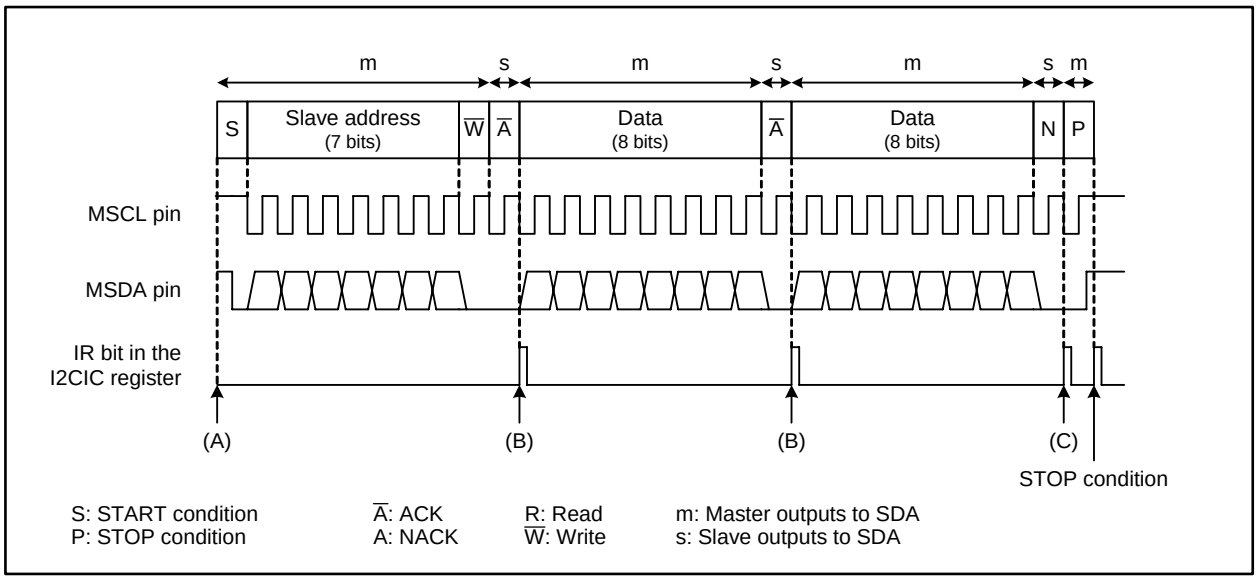


Figure 23.23 Example Operation of Master Transmission

A. Transmitting a slave address

(1) Confirm the BBSY bit in the I2CSR register is 0 (bus is free)
(2) Write E0h to the I2CSR register
→ The device enters the START condition standby state
(3) Write an address of a receiver (slave address) to the upper 7 bits of the I2CTRSR register
→ A START condition is generated
→ The slave address is sent

B. Transmitting data (processed in the I2C-bus interrupt routine)

(1) Write transmit data to the I2CTRSR register
→ Data is sent
To send multiple bytes of data, write them to the I2CTRSR register in succession

C. Completing master transmission (processed in the I2C-bus interrupt routine)

(1) Write C0h to the I2CSR register
→ The device enters the STOP condition standby state
(2) Write dummy data to the I2CTRSR register
→ A STOP condition is generated

In addition to the case where transmission is completed, procedure (C) is required when no ACK from the slave device is received (when a NACK is received as shown in Figure 23.23).

23.6.2 Slave Reception

The operation and procedures of slave reception are described in this section. Figure 23.24 shows an example of slave reception operation. For (A) to (D) in the figure, see A to D in the descriptions and procedures below. (1) to (3) show the program's instructions. Arrows indicate that the procedure is performed by the MCU automatically.

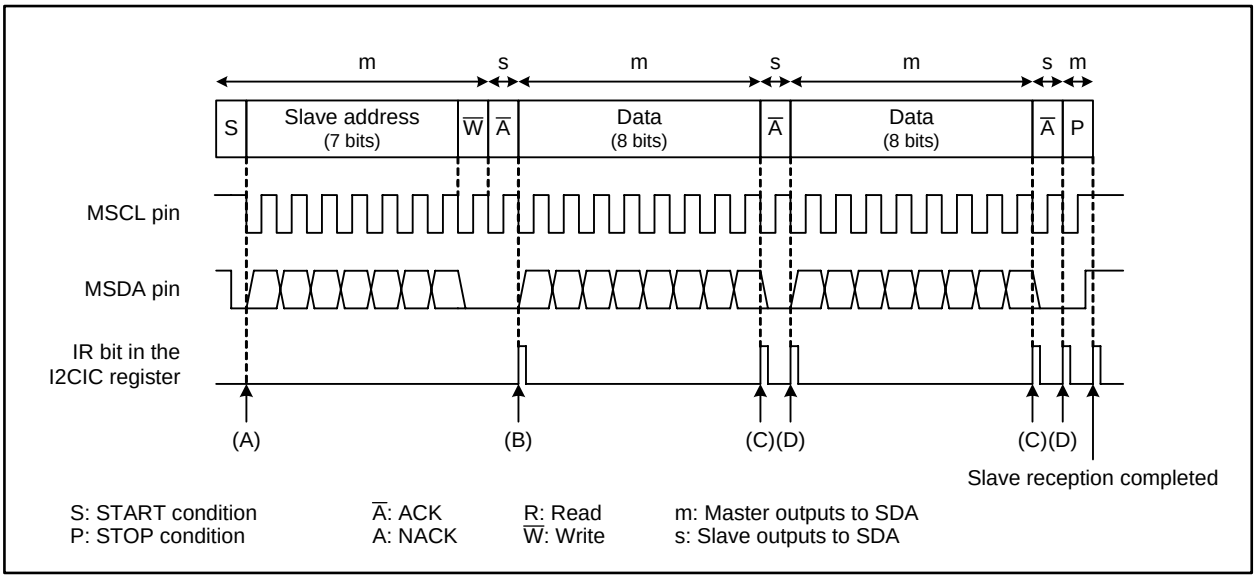


Figure 23.24 Example Operation of Slave Reception

A. Receiving a slave address (performed by the MCU automatically)
   → A START condition is detected
   → A slave address is received
   → An ACK is sent and the I2C-bus interface interrupt is generated in either of the following cases
   -When the general call address is received (the ADZ bit in the I2CSR register is 1)
   -When an address match is detected (the AAS bit in the I2CSR register is 1)

B. Starting slave reception (processed in the I2C-bus interrupt routine)
   (1) Confirm the content of the I2CSR register. When the TRS bit is 0, start the slave reception.
   (2) Write dummy data to the I2CTRSR register
   → Data reception starts

C. Completing slave reception (processed in the I2C-bus interrupt routine)
   (1) Read the received data from the I2CTRSR register
   (2) Set the ACKD bit in the register to 1 (NACK) when the data is the last received data
   (3) Set the ACKD bit in the register to 0 (ACK) when the data is not the last received data
   → An ACK or NACK is sent and an I2C-bus interface interrupt is generated

D. Completing ACK transmission (processed in the I2C-bus interrupt routine)
   (1) Write dummy data to the I2CTRSR register
   → If the data is the last received data, a STOP condition is detected
   → If not, data reception restarts

23.7 Notes on Using Multi-master I2C-bus Interface

23.7.1 Accessing Multi-master I2C-bus Interface-associated Registers

Notes on writing to and reading I2C-bus interface-associated registers.

- I2CTRSR register
  Do not write to this register during data transmission or reception. Doing so resets the transmit/receive counter and the register is unable to perform normal data transmission or reception.

- I2CCR0 register
  This register becomes 000b when a START condition is detected or 1 byte of data transmission or reception is completed. Do not write to or read this register at these two timings. Doing so may change the register value to an unexpected value. Figure 23.26 and Figure 23.27 show the bit counter reset timings.

- I2CCCR register
  Do not rewrite bits other than the ACKD bit during transmission or reception. Otherwise the I2C-bus clock circuit is reset and a normal transmission or reception will not be performed as a result.

- I2CCR1 register
  Rewrite bits ICK4 to ICK0 only when the ICE bit in the I2CCR0 register is 0 (I2C-bus interface disabled). When the I2CCR1 register is read, the internal WAIT flag status is read from this register. Therefore, do not use a bit processing instruction (read-modify-write instruction) with this register.

- I2CSR register
  Do not use a bit processing instruction (read-modify-write instruction) since the value of each bit in the I2CSR register changes depending on the communication state. Also, do not access this register when MST bit or TRS bit, which select the communication mode, changes. Doing so may change the register value to an unexpected value. Figure 23.25 to Figure 23.27 show the timing of bits MST and TRS to change.

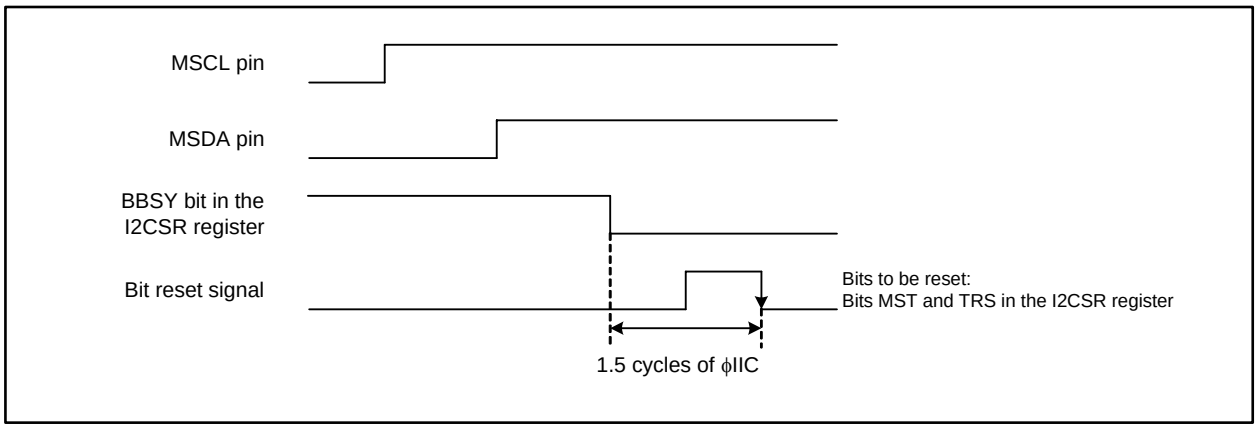

MSCL pin
MSDA pin
BBSY bit in the I2CSR register
Bit reset signal
Bits to be reset:
Bits MST and TRS in the I2CSR register
1.5 cycles of ϕIIC

Figure 23.25 Bit Resetting Timing (when a STOP condition is detected)

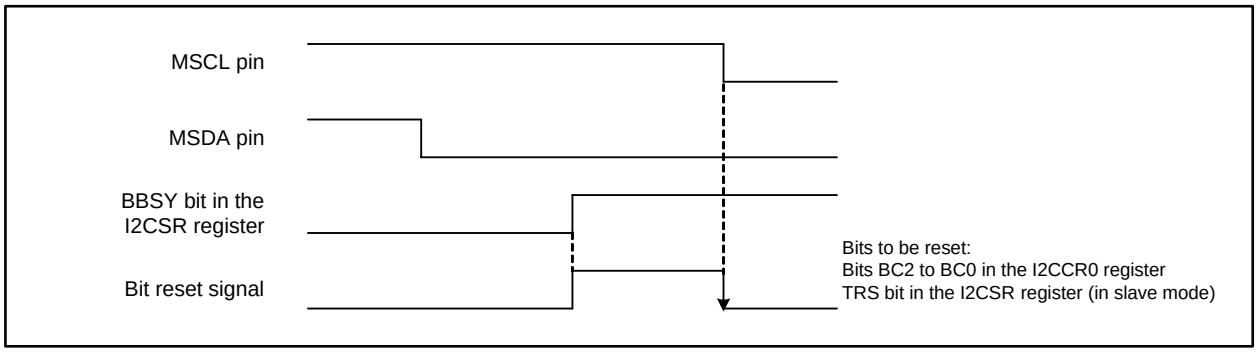

MSCL pin
MSDA pin
BBSY bit in the I2CSR register
Bit reset signal
Bits to be reset:
Bits BC2 to BC0 in the I2CCR0 register
TRS bit in the I2CSR register (in slave mode)

Figure 23.26 Bit Resetting Timing (when a START condition is detected)

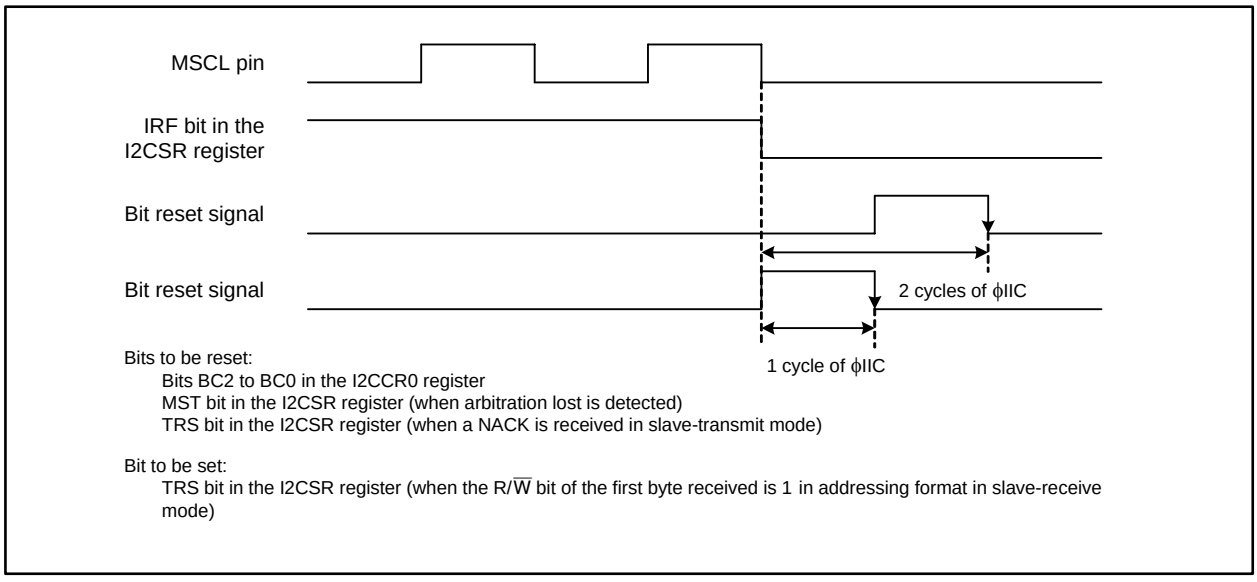

MSCL pin
IRF bit in the I2CSR register
Bit reset signal
Bit reset signal
2 cycles of ϕIIC
1 cycle of ϕIIC
Bits to be reset:
Bits BC2 to BC0 in the I2CCR0 register
MST bit in the I2CSR register (when arbitration lost is detected)
TRS bit in the I2CSR register (when a NACK is received in slave-transmit mode)
Bit to be set:
TRS bit in the I2CSR register (when the R/W bit of the first byte received is 1 in addressing format in slave-receive mode)

Figure 23.27 Bit Setting/Resetting Timing (when data transmission/reception is completed)

23.7.2 Generating a Repeated START condition

Use the following steps to generate a repeated START condition after transmitting 1-byte of data:

(1) Write E0h (the START condition standby state, and the MSDA pin is high) to the I2CSR register
(2) Wait until the MSDA pin becomes high
(3) Write a slave address to the I2CTRSR register to generate a START condition trigger

Figure 23.28 shows the repeated START condition generating timing.

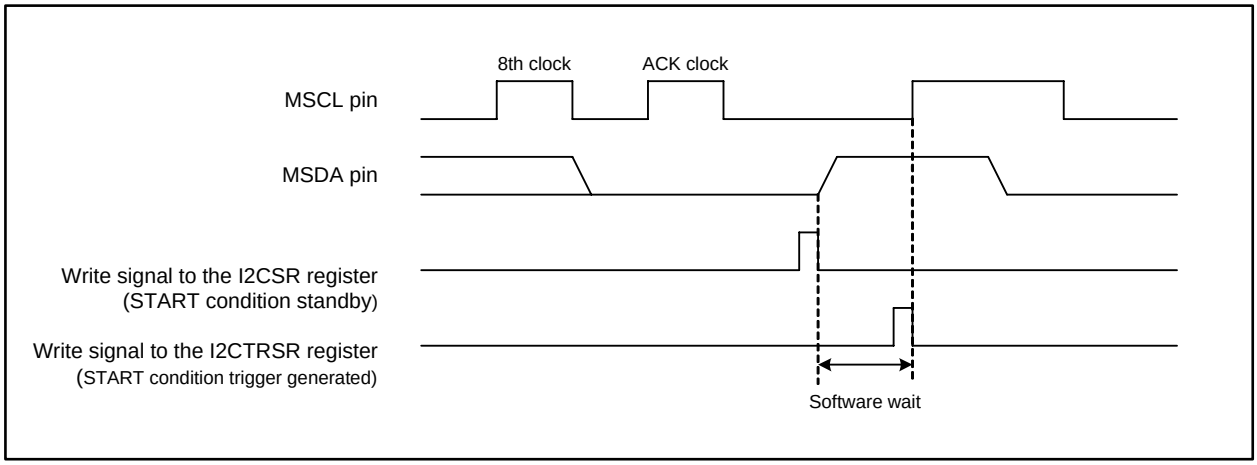


Figure 23.28 Repeated START Condition Generating Timing

24. CAN Module

The R32C/102 Group implements one channel (referred to as CAN0) of the Controller Area Network (CAN) module that complies with the ISO11898-1 Specifications. The CAN module transmits and receives both formats of messages, namely the standard identifier (11 bits) (identifier hereafter referred to as ID) and extended ID (29 bits).
Table 24.1 and Table 24.2 list the CAN module specifications, and Figure 24.1 shows the CAN module block diagram.
Connect the CAN bus transceiver externally.

Table 24.1 CAN Module Specifications (1)

Item	Specifications
Protocol	ISO11898-1 compliant
Bit rate	Up to 1 Mbps
Message boxes	32 mailboxes: Two selectable mailbox modes: • Normal mailbox mode All 32 mailboxes can be configured for transmission or reception. • FIFO mailbox mode: 24 mailboxes can be configured for transmission or reception. The remaining mailboxes can be configured as 4-stage FIFO for transmission and 4-stage FIFO for reception.
Reception	• Data frames and remote frames can be received. • Selectable receiving ID format (only standard ID, only extended ID, or both ID) • Programmable one-shot reception function • Selectable overwrite mode (message overwritten) or overrun mode (message discarded) • The reception complete interrupt can be individually enabled or disabled for each mailbox.
Acceptance filtering	8 acceptance masks: one mask every 4 mailboxes The mask can be individually enabled or disabled for each mailbox.
Transmission	• Data frame and remote frame can be transmitted. • Selectable transmitting ID format (only standard ID, only extended ID, or both ID). • Programmable one-shot transmission function • Selectable ID priority transmit mode or mailbox number priority transmit mode • Transmission request can be aborted. (The completion of abort can be confirmed with a flag.) • The transmission complete interrupt can be individually enabled or disabled for each mailbox.
Mode transition for bus-off recovery	Mode transition for the recovery from the bus-off state can be selected: • ISO11898-1 compliant • Automatic entry to CAN halt mode at bus-off entry • Automatic entry to CAN halt mode at bus-off end • Entry to CAN halt mode by a program • Transition to the error-active state by a program

Table 24.2 CAN Module Specifications (2)

Item	Specifications
Error status monitoring	• CAN bus errors (stuff error, form error, ACK error, CRC error, bit error, and ACK delimiter error) can be monitored. • Transition to error states can be detected (error-warning, error-passive, bus-off entry, and bus-off recovery). • The error counters can be read.
Time stamp function	Time stamp function using a 16-bit counter The reference clock can be selected from either 1-, 2-, 4- or 8-bit time periods.
Interrupt sources	6 types: • Reception complete • Transmission complete • Receive FIFO • Transmit FIFO • Error • Wake-up
CAN sleep mode	Current consumption can be reduced by stopping the CAN clock.
Software support units	3 software support units: • Acceptance filter support • Mailbox search support (receive mailbox search, transmit mailbox search, and message lost search) • Channel search support
CAN clock source	Selectable peripheral bus clock or main clock
Test mode	3 test modes available for user evaluation: • Listen only mode • Self test mode 0 (external loop back) • Self test mode 1 (internal loop back)

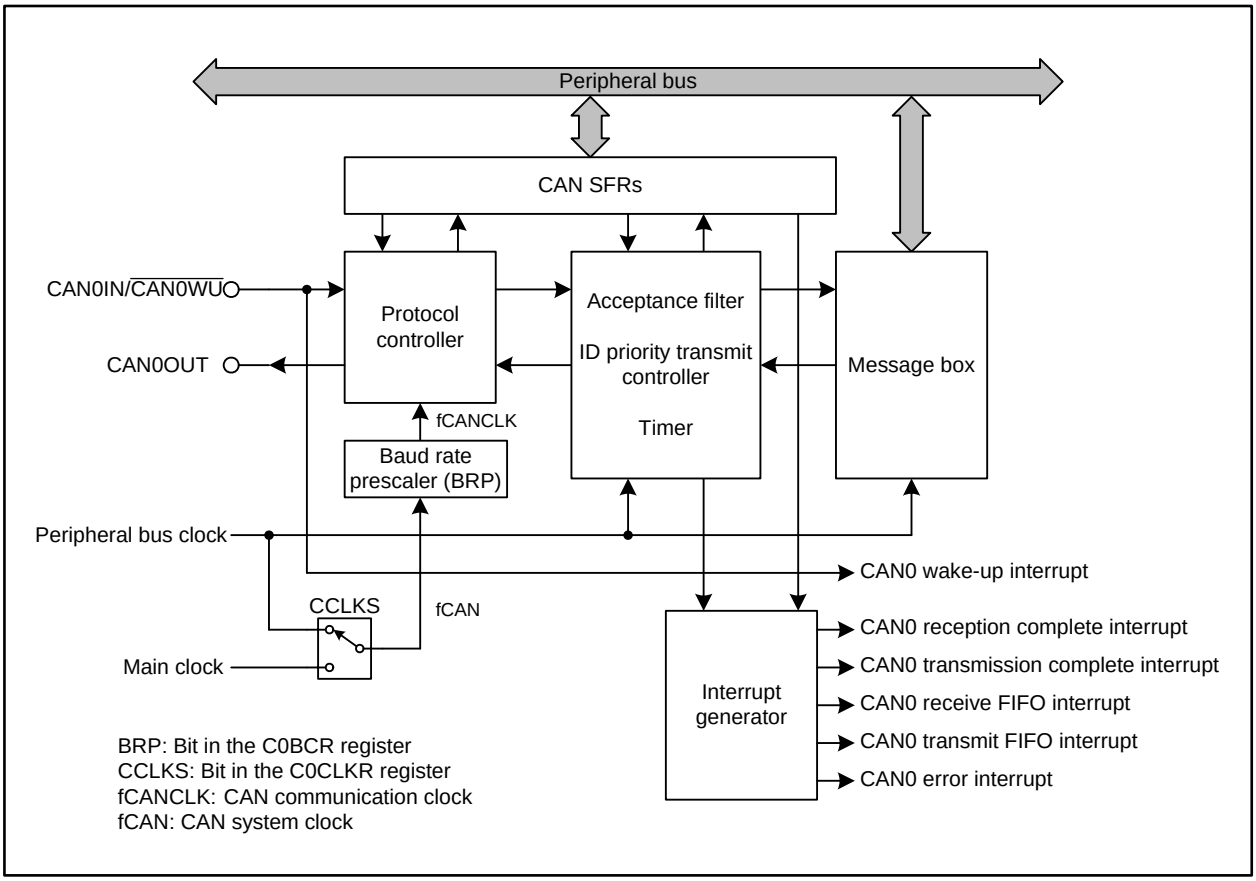


Figure 24.1 CAN Module Block Diagram

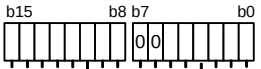
- CAN0IN/CAN0OUT: CAN input/output pins
- Protocol controller: Handles CAN protocol processing such as bus arbitration, bit timing at transmission and reception, stuffing, and error handling, etc.
- Message box: Consists of 32 mailboxes which can be configured as either transmit or receive mailboxes. Each mailbox has an individual ID, data length code, a data field (8 bytes), and a time stamp.
- Acceptance filter: Performs filtering of received messages. Registers C0MKR0 to C0MKR7 are used for the filtering process.
- Timer: Used for the time stamp function. The timer value when storing a message into the mailbox is written as the time stamp value.
- Wake-up function: Generates a CAN0 wake-up interrupt request when a message is detected on the CAN bus.
- Interrupt generator: Generates the following five types of interrupts:
  - CAN0 reception complete interrupt
  - CAN0 transmission complete interrupt
  - CAN0 receive FIFO interrupt
  - CAN0 transmit FIFO interrupt
  - CAN0 error interrupt
- CAN SFRs: CAN-associated registers. Refer to 24.1 “CAN SFRs” for details.

24.1 CAN SFRs

The CAN-associated registers are shown in Figures 24.2 to 24.11, 24.13, 24.14, 24.16 to 24.20, 24.22, and 24.24 to 24.30.

24.1.1 CAN0 Control Register (C0CTLR Register)

CAN0 Control Register



Symbol	Address	Reset Value
C0CTLR	47F41h-47F40h	0000 0000 0000 0101b

Bit Symbol	Bit Name	Function	RW
CANM	CAN Operating Mode Select Bit (1)	b1 b0 0 0 : CAN operation mode 0 1 : CAN reset mode 1 0 : CAN halt mode 1 1 : Do not use this combination	RW
SLPM	CAN Sleep Mode Bit (1, 2)	0: Other than CAN sleep mode 1: CAN sleep mode	RW
BOM	Bus-off Recovery Mode Select Bit (3)	b4 b3 0 0 : Normal mode (ISO11898-1 compliant) 0 1: Entry to CAN halt mode automatically at bus-off entry 1 0 : Entry to CAN halt mode automatically at bus-off end 1 1 : Entry to CAN halt mode (during bus-off recovery period) by a program request	RW
RBOC	Forcible Return From Bus-off Bit (4)	0: Nothing occurred 1: Forcible return from bus-off (5)	RW
— (b7-b6)	Reserved	Should be written with 0	RW
MBM	CAN Mailbox Mode Select Bit (3)	0: Normal mailbox mode 1: FIFO mailbox mode	RW
IDFM	ID Format Mode Select Bit (3)	b10b9 0 0 : Standard ID mode 0 1 : Extended ID mode 1 0 : Mixed ID mode 1 1 : Do not use this combination	RW
MLM	Message Lost Mode Select Bit (3)	0: Overwrite mode 1: Overrun mode	RW
TPM	Transmit Priority Mode Select Bit (3)	0: ID priority transmit mode 1: Mailbox number priority transmit mode	RW
TSRC	Time Stamp Counter Reset Bit (6)	0: Nothing occurred 1: Reset (5)	RW
TSPS	Time Stamp Prescaler Select Bit (3)	b15b14 0 0 : Every bit time 0 1 : Every 2-bit time 1 0 : Every 4-bit time 1 1 : Every 8-bit time	RW

Notes:
1. When bits CANM and SLPM are changed, check the C0STR register to ensure that the mode has been switched.
   Do not change bits CANM and SLPM until the mode has been switched.
2. Write to the SLPM bit in CAN reset mode or CAN halt mode.
   When rewriting the SLPM bit, set only this bit to 0 or 1.
3. Write to bits BOM, MBM, IDFM, MLM, TPM, and TSPS in CAN reset mode.
4. Set the RBOC bit to 1 in bus-off state.
5. Bits RBOC and TSRC are automatically set back to 0 after being set to 1. They are read as 0.
6. Set the TSRC bit to 1 in CAN operation mode.

Figure 24.2 C0CTLR Register

24.1.1.1 CANM Bit

The CANM bit selects one of the following modes for the CAN module: CAN operation mode, CAN reset mode, or CAN halt mode. Refer to 24.2 “Operating Mode” for detail.
CAN sleep mode is set by the SLPM bit.
Do not set the CANM bit to 11b.
When the CAN module enters CAN halt mode according to the setting of the BOM bit, the CANM bit is automatically set to 10b.

24.1.1.2 SLPM Bit

When the SLPM bit is set to 1, the CAN module enters CAN sleep mode.
When this bit is set to 0, the CAN module exits CAN sleep mode.
Refer to 24.2 “Operating Mode” for detail.

24.1.1.3 BOM Bit

The BOM bit is used to select bus-off recovery mode.

When the BOM bit is 00b, the recovery from bus-off is compliant with ISO11898-1, i.e. the CAN module re-enters CAN communication (error-active state) after detecting 11 consecutive recessive bits 128 times. A bus-off recovery interrupt request is generated when recovering from bus-off.

When the BOM bit is 01b, as soon as the CAN module reaches the bus-off state, the CANM bit in the C0CTLR register is set to 10b (CAN halt mode) and the CAN module enters CAN halt mode. No bus-off recovery interrupt request is generated when recovering from bus-off and registers C0TECR and C0RECR are set to 00h.

When the BOM bit is 10b, the CANM bit is set to 10b as soon as the CAN module reaches the bus-off state. The CAN module enters CAN halt mode after the recovery from the bus-off state, i.e. after detecting 11 consecutive recessive bits 128 times. A bus-off recovery interrupt request is generated when recovering from bus-off and registers C0TECR and C0RECR are set to 00h.

When the BOM bit is 11b, the CAN module enters CAN halt mode by setting the CANM bit to 10b while the CAN module is still in bus-off state. No bus-off recovery interrupt request is generated when recovering from bus-off and registers C0TECR and C0RECR are set to 00h. However, if the CAN module recovers from bus-off after detecting 11 consecutive recessive bits 128 times before the CANM bit is set to10b, a bus-off recovery interrupt request is generated.

If the CPU requests an entry to CAN reset mode at the same time as the CAN module attempts to enter CAN halt mode (at bus-off entry when the BOM bit is 01b, or at bus-off end when the BOM bit is 10b), then the CPU request to enter CAN reset mode has higher priority.

24.1.1.4 RBOC Bit

When the RBOC bit is set to 1 (forcible return from bus-off) in bus-off state, the CAN module forcibly returns from the bus-off state. This bit is automatically set to 0. The error state changes from bus-off to error-active.

When the RBOC bit is set to 1, registers C0RECR and C0TECR are set to 00h and the BOST bit in the C0STR register is set to 0 (the CAN module is not in bus-off state). The other registers remain unchanged. No bus-off recovery interrupt request is generated by this recovery from the bus-off state.
Use the RBOC bit only when the BOM bit is 00b (normal mode).

24.1.1.5 MBM Bit

When the MBM bit is 0 (normal mailbox mode), mailboxes [0] to [31] are configured as transmit or receive mailboxes.
When this bit is 1 (FIFO mailbox mode), mailboxes [0] to [23] are configured as transmit or receive mailboxes. Mailboxes [24] to [27] are configured as a transmit FIFO and mailboxes [28] to [31] as a receive FIFO.
Transmit data is written into mailbox [24] (mailbox [24] is a window mailbox for the transmit FIFO).
Receive data is read from mailbox [28] (mailbox [28] is a window mailbox for the receive FIFO).
Table 24.3 lists the mailbox configuration.

Table 24.3 Mailbox Configuration

Mailbox	MBM = 0 (Normal Mailbox Mode)	MBM = 1 (1) (FIFO Mailbox Mode)
Mailboxes [0] to [23]	Normal mailbox	Normal mailbox
Mailboxes [24] to [27]		Transmit FIFO
Mailboxes [28] to [31]		Receive FIFO

Note:
1. When the MBM bit is set to 1, note the following:
   - Transmit FIFO is controlled by the C0TFCR register.
     The C0MCTLj register (j = 0 to 31) for mailboxes [24] to [27] is disabled.
     Registers C0MCTL24 to C0MCTL27 cannot be used.
   - Receive FIFO is controlled by the C0RFCR register.
     The C0MCTLj register for mailboxes [28] to [31] is disabled.
     Registers C0MCTL28 to C0MCTL31 cannot be used.
   - Refer to the C0MIER register about the FIFO interrupts.
   - The corresponding bits in the C0MKIVLR register for mailboxes [24] to [31] are disabled. Set 0 to these bits.
   - Transmit/receive FIFOs can be used for both data frames and remote frames.

24.1.1.6 IDFM Bit

The IDFM bit specifies the ID format.
When this bit is 00b, all mailboxes (including FIFO mailboxes) handle only standard IDs.
When this bit is 01b, all mailboxes (including FIFO mailboxes) handle only extended IDs.
When this bit is 10b, all mailboxes (including FIFO mailboxes) handle both standard IDs and extended IDs. Standard IDs or extended IDs are specified by using the IDE bit in the corresponding mailbox in normal mailbox mode. In FIFO mailbox mode, the IDE bit in the corresponding mailbox is used for mailboxes [0] to [23], the IDE bit in registers C0FIDCR0 and C0FIDCR1 is used for the receive FIFO, and the IDE bit in mailbox [24] is used for the transmit FIFO.
Do not set 11b to the IDFM bit.

24.1.1.7 MLM Bit

The MLM bit specifies the operation when a new message is captured in the unread mailbox.
Overwrite mode or overrun mode can be selected. All mailboxes (including the receive FIFO) are set to either overwrite mode or overrun mode.

When the MLM bit is 0, all mailboxes are set to overwrite mode and the new message is overwriting the old message.
When this bit is 1, all mailboxes are set to overrun mode and the new message is discarded.

24.1.1.8 TPM Bit

The TPM bit specifies the priority of modes when transmitting messages.
ID priority transmit mode or mailbox number transmit mode can be selected.
All mailboxes are set for either ID priority transmission or mailbox number priority transmission.

When the TPM bit is 0, ID priority transmit mode is selected and transmission priority complies with the CAN bus arbitration rule, as defined in the ISO 11898-1 Specifications. In ID priority transmit mode, mailboxes [0] to [31] (in normal mailbox mode), and mailboxes [0] to [23] (in FIFO mailbox mode), and the transmit FIFO are compared for the IDs of mailboxes configured for transmission. If two or more mailbox IDs are the same, the mailbox with the smaller number has higher priority.

Only the next message to be transmitted from the transmit FIFO is included in the transmission arbitration. If a transmit FIFO message is being transmitted, the next pending message within the transmit FIFO is included in the transmission arbitration.

When the TPM bit is 1, mailbox number transmit mode is selected and the transmit mailbox with the smallest mailbox number has the highest priority. In FIFO mailbox mode, the transmit FIFO has lower priority than normal mailboxes (mailboxes [0] to [23]).

24.1.1.9 TSRC Bit

The TSRC bit is used to reset the time stamp counter.
When this bit is set to 1, the C0TSR register is set to 0000h. It is automatically set to 0.

24.1.1.10 TSPS Bit

The TSPS bit selects the prescaler for the time stamp.
The reference clock for the time stamp can be selected from either 1-, 2-, 4- or 8-bit time periods.

24.1.2 CAN0 Clock Select Register (C0CLKR Register)

CAN0 Clock Select Register

b7	b6	b5	b4	b3	b2	b1	b0
0	X	X	0	0	X	0	

Symbol	Address	Reset Value
C0CLKR	47F47h	000X 0000b

Bit Symbol	Bit Name	Function	RW
CCLKS	CAN Clock Source Select Bit (1)	0: Peripheral bus clock 1: Main clock (2)	RW
— (b1)	Reserved	Should be written with 0	RW
— (b2)	No register bit; should be written with 0 and read as 0		—
— (b3)	Reserved	Should be written with 0	RW
— (b4)	Reserved	Should be written with 0 and read as undefined value	RW
— (b6-b5)	No register bits; should be written with 0 and read as 0		—
— (b7)	Reserved	Should be written with 0	RW

Notes:
1. Write to the CCLKS bit in CAN reset mode.
2. To set the CCLKS bit to 1, the frequency of the peripheral bus clock should be equal to or higher than the frequency of the main clock.

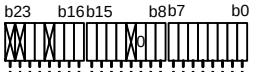
Figure 24.3 C0CLKR Register

24.1.2.1 CCLKS Bit

When the CCLKS bit is set to 0, the CAN clock source (fCAN) originates from the PLL.
When this bit is set to 1, the fCAN originates directly from the external XIN pin bypassing the PLL.

24.1.3 CAN0 Bit Configuration Register (C0BCR Register)

CAN0 Bit Configuration Register (1, 2)



Symbol	Address	Reset Value
C0BCR	47F46h-47F44h	00 0000h

Bit Symbol	Bit Name	Function	RW
BRP	Prescaler Division Ratio Set Bit (10 bits)	If the setting value is P (0 to 1023), the baud rate prescaler divides fCAN by P + 1	RW
— (b10)	Reserved	Should be written with 0	RW
— (b11)	No register bit; should be written with 0 and read as 0		—
TSEG1	Time Segment 1 Control Bit	b15b14b13b12 0 0 0 0 : Do not use this combination 0 0 0 1 : Do not use this combination 0 0 1 0 : Do not use this combination 0 0 1 1 : 4 Tq 0 1 0 0 : 5 Tq 0 1 0 1 : 6 Tq 0 1 1 0 : 7 Tq 0 1 1 1 : 8 Tq 1 0 0 0 : 9 Tq 1 0 0 1 : 10 Tq 1 0 1 0 : 11 Tq 1 0 1 1 : 12 Tq 1 1 0 0 : 13 Tq 1 1 0 1 : 14 Tq 1 1 1 0 : 15 Tq 1 1 1 1 : 16 Tq	RW
TSEG2	Time Segment 2 Control Bit	b18b17b16 0 0 0 : Do not use this combination 0 0 1 : 2 Tq 0 1 0 : 3 Tq 0 1 1 : 4 Tq 1 0 0 : 5 Tq 1 0 1 : 6 Tq 1 1 0 : 7 Tq 1 1 1 : 8 Tq	RW
— (b19)	No register bit; should be written with 0 and read as 0		—
SJW	Resynchronization Jump Width Control Bit	b21b20 0 0 : 1 Tq 0 1 : 2 Tq 1 0 : 3 Tq 1 1 : 4 Tq	RW
— (b23-b22)	No register bits; should be written with 0 and read as 0		—

Notes:
1. Set the C0BCR register before entering CAN halt mode from CAN reset mode or CAN operation mode from CAN reset mode. After the setting is made once, this register can be written to in CAN reset mode or CAN halt mode.
2. The C0BCR register consists of 24 bits. A 32-bit read/write access should be performed carefully not to rewrite the C0CLKR register.

Figure 24.4 C0BCR Register

Refer to 24.3 “CAN Communication Speed Configuration” about the bit timing configuration.

24.1.3.1 BRP Bit

The BRP bit is used to set the frequency of the CAN communication clock (fCANCLK).
The cycle of the fCANCLK is set to be 1 Time Quantum (Tq).

24.1.3.2 TSEG1 Bit

The TSEG1 bit is used to specify the total length of the propagation time segment (PROP_SEG) and phase buffer segment 1 (PHASE_SEG1) with the value of Tq.
A value from 4 to 16 time quanta can be set.

24.1.3.3 TSEG2 Bit

The TSEG2 bit is used to specify the length of phase buffer segment TSEG2 (PHASE_SEG2) with the value of Tq.
A value from 2 to 8 time quanta can be set.
Set the value smaller than that of the TSEG1 bit.

24.1.3.4 SJW Bit

The SJW bit is used to specify the resynchronization jump width with the value of Tq.
A value from 1 to 4 time quanta can be set.
Set the value smaller than or equal to that of the TSEG2 bit.

24.1.4 CAN0 Mask Register k (C0MKRk Register) (k = 0 to 7)

CAN0 Mask Register k (k = 0 to 7) (1)



Symbol	Address	Reset Value
C0MKR0, C0MKR1	47E03h-47E00h, 47E07h-47E04h	Undefined
C0MKR2, C0MKR3	47E0Bh-47E08h, 47E0Fh-47E0Ch	Undefined
C0MKR4, C0MKR5	47E13h-47E10h, 47E17h-47E14h	Undefined
C0MKR6, C0MKR7	47E1Bh-47E18h, 47E1Fh-47E1Ch	Undefined

Bit Symbol	Bit Name	Function	RW
EID	Extended ID Bit	0: Corresponding EID bit is not compared 1: Corresponding EID bit is compared	RW
SID	Standard ID Bit	0: Corresponding SID bit is not compared 1: Corresponding SID bit is compared	RW
— (b31-b29)	Reserved	Should be written with 0	RW

Note:
1. Write to registers C0MKR0 to C0MKR7 in CAN reset mode or CAN halt mode.

Figure 24.5 Registers C0MKR0 to C0MKR7

Refer to 24.5 “Acceptance Filtering and Masking Function” about the masking function in FIFO mailbox mode.

24.1.4.1 EID Bit

The EID bit is the filter mask bit corresponding to the CAN extended ID bit. This bit is used to receive extended ID messages.
When the EID bit is 0, the corresponding EID bit is not compared for the received ID and the mailbox ID.
When this bit is 1, the corresponding EID bit is compared for the received ID and the mailbox ID.

24.1.4.2 SID Bit

The SID bit is the filter mask bit corresponding to the CAN standard ID bit. This bit is used to receive both standard ID and extended ID messages.
When the SID bit is 0, the corresponding SID bit is not compared for the received ID and the mailbox ID.
When this bit is 1, the corresponding SID bit is compared for the received ID and the mailbox ID.

24.1.5 CAN0 FIFO Received ID Compare Register n (Registers C0FIDCR0 and C0FIDCR1) (n = 0, 1)

CAN0 FIFO Received ID Compare Register n (n = 0, 1) (1)



Symbol	Address	Reset Value
C0FIDCR0	47E23h-47E20h	Undefined
C0FIDCR1	47E27h-47E24h	Undefined

Bit Symbol	Bit Name	Function	RW
EID	Extended ID Bit	0: Corresponding EID bit is 0 1: Corresponding EID bit is 1	RW
SID	Standard ID Bit	0: Corresponding SID bit is 0 1: Corresponding SID bit is 1	RW
— (b29)	Reserved	Should be written with 0	RW
RTR	Remote Frame Request Bit	0: Data frame 1: Remote frame	RW
IDE	ID Extension Bit (2)	0: Standard ID 1: Extended ID	RW

Notes:
1. Write to registers C0FIDCR0 and C0FIDCR1 in CAN reset mode or CAN halt mode.
2. The IDE bit is enabled when the IDFM bit in the C0CTLR register is 10b (mixed ID mode). When the IDFM bit is either 00b (standard ID mode) or 01b (extended ID mode), the IDE bit should be written with 0.

Figure 24.6 Registers C0FIDCR0 and C0FIDCR1

Registers C0FIDCR0 and C0FIDCR1 are enabled when the MBM bit in the C0CTLR register is set to 1 (FIFO mailbox mode). Bits EID, SID, RTR, and IDE in registers C0MB28 to C0MB31 are disabled. Refer to 24.5 “Acceptance Filtering and Masking Function” about the usage of these registers.

24.1.5.1 EID Bit

The EID bit sets the extended ID of data frames and remote frames. This bit is used to receive extended ID messages.

24.1.5.2 SID Bit

The SID bit sets the standard ID of data frames and remote frames. This bit is used to receive both standard ID and extended ID messages.

24.1.5.3 RTR Bit

The RTR bit sets the specified frame format of data frames or remote frames.
This bit specifies the following operation:

- When both RTR bits in registers C0FIDCR0 and C0FIDCR1 are set to 0, only data frames can be received.
- When both RTR bits in registers C0FIDCR0 and C0FIDCR1 are set to 1, only remote frames can be received.
- When the RTR bits in registers C0FIDCR0 and C0FIDCR1 are set to 0 or 1 individually, both data frames and remote frames can be received.

24.1.5.4 IDE Bit

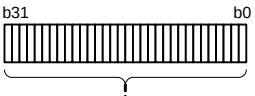
The IDE bit sets the ID format of standard ID or extended ID.
This bit is enabled when the IDFM bit in the C0CTLR register is 10b (mixed ID mode).
When the IDFM bit is 10b, the IDE bit specifies the following operation:

- When both IDE bits in registers C0FIDCR0 and C0FIDCR1 are set to 0, only standard ID frames can be received.
- When both IDE bits in registers C0FIDCR0 and C0FIDCR1 are set to 1, only extended ID frames can be received.
- When the IDE bits in registers C0FIDCR0 and C0FIDCR1 are set to 0 or 1 individually, both standard ID and extended ID frames can be received.

24.1.6 CAN0 Mask Invalid Register (C0MKIVLR Register)

CAN0 Mask Invalid Register (1)

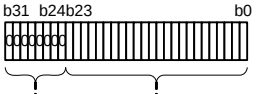
Normal mailbox mode



Symbol	Address	Reset Value
C0MKIVLR	47E2Bh-47E28h	Undefined

Bit Symbol	Bit Name	Function	RW
— (b31-b0)	Mask Invalid Bit	0: Mask valid 1: Mask invalid	RW

FIFO mailbox mode



Bit Symbol	Bit Name	Function	RW
— (b23-b0)	Mask Invalid Bit	0: Mask valid 1: Mask invalid	RW
— (b31-b24)	Reserved	Should be written with 0	RW

Note:
1. Write to the C0MKIVLR register in CAN reset mode or CAN halt mode.

Figure 24.7 C0MKIVLR Register

Each bit corresponds to the mailbox with the same number. When each bit is 1, the acceptance mask for the mailbox corresponding to the bit number is disabled. In this case, a receiving message is stored into the mailbox only if its ID matches bits SID and EID in the C0MBj register (j = 0 to 31).

24.1.7 CAN0 Mailbox (C0MBj Register) (j = 0 to 31)

Table 24.4 lists the CAN0 mailbox memory mapping, and Table 24.5 lists the CAN data frame structure. The value after reset of CAN0 mailbox is undefined.

Table 24.4 CAN0 Mailbox Memory Mapping

Address	Message Content
CAN0	Memory Mapping
47C00h + j × 16 + 0	EID7 to EID0
47C00h + j × 16 + 1	EID15 to EID8
47C00h + j × 16 + 2	SID5 to SID0, EID17, EID16
47C00h + j × 16 + 3	IDE, RTR, SID10 to SID6
47C00h + j × 16 + 4	—
47C00h + j × 16 + 5	Data length code (DLC)
47C00h + j × 16 + 6	Data byte 0
47C00h + j × 16 + 7 : : : 47C00h + j × 16 + 13	Data byte 1 : : : Data byte 7
47C00h + j × 16 + 14	Time stamp lower byte
47C00h + j × 16 + 15	Time stamp upper byte

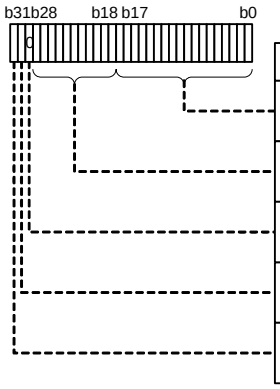
j: Mailbox number (j = 0 to 31)

Table 24.5 CAN Data Frame Structure

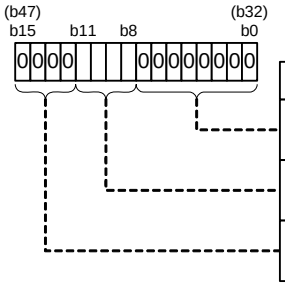
SID10 to SID6	SID5 to SID0	EID17 to EID16	EID15 to EID8	EID7 to EID0	DLC3 to DLC0	DATA0	DATA1	.....	DATA7

CAN0 Mailbox Register j (j = 0 to 31) (1)

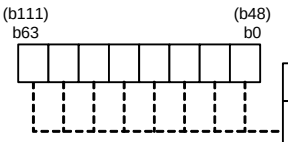
Symbol	Address (2)	Reset Value
C0MB0 to C0MB31	47C00h to 47DFFh	Undefined



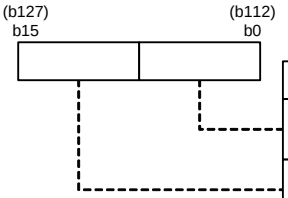
Bit Symbol	Bit Name	Function	RW
EID	Extended ID (3)	0: Corresponding EID bit is 0 1: Corresponding EID bit is 1	RW
SID	Standard ID	0: Corresponding SID bit is 0 1: Corresponding SID bit is 1	RW
— (b29)	Reserved	Should be written with 0	RW
RTR	Remote Frame Request Bit	0: Data frame 1: Remote frame	RW
IDE	ID Extension Bit (4)	0: Standard ID 1: Extended ID	RW



Bit Symbol	Bit Name	Setting Range	RW
— (b7-b0)	Reserved	Should be written with 0	RW
DLC	Data Length Code (5)	0h to Fh	RW
— (b15-b12)	Reserved	Should be written with 0	RW



Symbol	Name	Setting Range	RW
DATA0 to DATA7	Data Bytes 0 to 7 (5, 6)	00h to FFh	RW



Symbol	Name	Setting Range	RW
TSL	Time Stamp Lower Byte	00h to FFh	RW
TSH	Time Stamp Higher Byte	00h to FFh	RW

Notes:
1. Write to the C0MBj register only when the associated C0MCTLj register is 00h and the corresponding mailbox is not processing an abort request.
2. Refer to the memory mapping table for CAN0 mailbox on the previous page for detailed addresses.
3. If the mailbox has received a standard ID message, the EID bit in the mailbox is undefined.
4. The IDE bit is enabled when the IDFM bit in the C0CTLR register is 10b (mixed ID mode). When the IDFM bit is either 00b (standard ID mod) or 01b (extended ID mode), it should be written with 0.
5. If the mailbox has received a message with n bytes less than 8 bytes, the values of DATAn to DATA7 in the mailbox are undefined.
6. If the mailbox has received a remote frame, the previous values of DATA0 to DATA7 in the mailbox are retained.

Figure 24.8 C0MBj Register

The previous value of each mailbox is retained unless a new message is received.

24.1.7.1 EID Bit

The EID bit sets the extended ID of data frames and remote frames. This bit is used to transmit or receive extended ID messages.

24.1.7.2 SID Bit

The SID bit sets the standard ID of data frames and remote frames. This bit is used to transmit or receive both standard ID and extended ID messages.

24.1.7.3 RTR Bit

The RTR bit sets the frame format of data frames or remote frames.
This bit specifies the following operation:

- Receive mailbox receives only frames with the format specified by the RTR bit.
- Transmit mailbox transmits according to the frame format specified by the RTR bit.
- Receive FIFO mailbox receives the data frame, remote frame, or both frames specified by the RTR bit in registers C0FIDCR0 and C0FIDCR1.
- Transmit FIFO mailbox transmits the data frame or remote frame specified by the RTR bit in the relevant transmitting message.

24.1.7.4 IDE Bit

The IDE bit sets the ID format of standard IDs or extended IDs.
This bit is enabled when the IDFM bit in the C0CTLR register is 10b (mixed ID mode).
When the IDFM bit is 10b, the IDE bit specifies the following operation:

- Receive mailbox receives only the ID format specified by the IDE bit.
- Transmit mailbox transmits according to the ID format specified by the IDE bit.
- Receive FIFO mailbox receives messages with the standard ID, extended ID, or both IDs specified by the IDE bit in registers C0FIDCR0 and C0FIDCR1.
- Transmit FIFO mailbox transmits messages with the standard ID or extended ID specified by the IDE bit in the relevant transmitting message.

24.1.7.5 DLC (Data Length Code)

The DLC is used to set the number of data bytes to be transmitted in a data frame. When data is requested using a remote frame, the number of data bytes to be requested is set.
When a data frame is received, the number of received data bytes is stored. When a remote frame is received, the number of requested data bytes is stored.
Table 24.6 lists the data length corresponding DLC.

Table 24.6 Data Length Corresponding DLC

DLC[3]	DLC[2]	DLC[1]	DLC[0]	Data Length
0	0	0	0	0 byte
0	0	0	1	1 byte
0	0	1	0	2 bytes
0	0	1	1	3 bytes
0	1	0	0	4 bytes
0	1	0	1	5 bytes
0	1	1	0	6 bytes
0	1	1	1	7 bytes
1	X	X	X	8 bytes

X: Any value

24.1.7.6 DATA0 to DATA7

DATA0 to DATA7 store the transmitted or received CAN message data. Transmission or reception starts from DATA0. The bit order on the CAN bus is MSB first, and transmission or reception starts from bit 7.

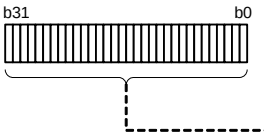
24.1.7.7 TSL and TSH

TSL and TSH store the counter value of the time stamp when received messages are stored in the mailbox.

24.1.8 CAN0 Mailbox Interrupt Enable Register (C0MIER Register)

CAN0 Mailbox Interrupt Enable Register (1, 2)

Normal mailbox mode



Symbol	Address	Reset Value
C0MIER	47E2Fh-47E2Ch	Undefined

Bit Symbol	Bit Name	Function	RW
— (b31-b0)	Interrupt Enable Bit	0: Interrupt disabled 1: Interrupt enabled	RW

FIFO mailbox mode



Bit Symbol	Bit Name	Function	RW
— (b23-b0)	Interrupt Enable Bit	0: Interrupt disabled 1: Interrupt enabled	RW
— (b24)	Transmit FIFO Interrupt Enable Bit	0: Interrupt disabled 1: Interrupt enabled	RW
— (b25)	Transmit FIFO Interrupt Generation Timing Control Bit	Transmit FIFO interrupt request is generated 0: Every time transmission is completed 1: When transmit FIFO becomes empty due to completion of transmission	RW
— (b27-b26)	Reserved	Should be written with 0	RW
— (b28)	Receive FIFO Interrupt Enable Bit	0: Interrupt disabled 1: Interrupt enabled	RW
— (b29)	Receive FIFO Interrupt Generation Timing Control Bit	Receive FIFO interrupt request is generated 0: Every time reception is completed 1: When receive FIFO becomes buffer warning by completion of reception (3)	RW
— (b31-b30)	Reserved	Should be written with 0	RW

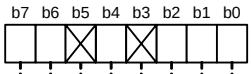
Notes:
1. Write to the C0MIER register only when the associated C0MCTLj register (j = 0 to 31) is 00h and the corresponding mailbox is not processing a transmission or reception abort request.
2. In FIFO mailbox mode, change the bits in the C0MIER register for the associated FIFO only when:
   - The TFE bit in the C0TFCR register is 0 and the TFEST bit is 1, and
   - The RFE bit in the C0RFCR register is 0 and the RFEST bit is 1.
3. No interrupt request is generated when the receive FIFO becomes buffer warning from full.

Figure 24.9 C0MIER Register

Interrupts can enabled individually for each mailbox.
In normal mailbox mode (bits 0 to 31) and in FIFO mailbox mode (bits 0 to 23), each bit corresponds to the mailbox with the same number. These bits enable or disable transmission/reception complete interrupts for the corresponding mailboxes.
In FIFO mailbox mode, bits 24, 25, 28, and 29 specify whether transmit/receive FIFO interrupts are enabled/disabled and timing when interrupt requests are generated.
“Buffer warning” indicates a state in which the third unread message is stored in the receive FIFO.

24.1.9 CAN0 Message Control Register j (C0MCTLj Register) (j = 0 to 31)

CAN0 Message Control Register j (j = 0 to 31) (1, 2)



Symbol	Address	Reset Value
C0MCTL0 to C0MCTL31	47F20h to 47F3Fh	00h

Bit Symbol	Bit Name	Function	RW
When the TRMREQ bit is 0 and the RECREQ bit is 1			
NEWDATA	Reception Complete Flag (3, 4)	0: No data has been received or 0 is written to the NEWDATA bit 1: A new message is being stored or has been stored to the mailbox	RW
INVALDATA	Reception-in-progress Status Flag	0: Message valid 1: Message being updated	RO
MSGLOST	Message Lost Flag (3, 4)	0: Message is not overwritten or overrun 1: Message is overwritten or overrun	RW
When the TRMREQ bit is 1 and the RECREQ bit is 0			
SENTDATA	Transmission Complete Flag (3, 4)	0: Transmission is not completed (pending) 1: Transmission is completed (success)	RW
TRMACTIVE	Transmission-in-progress Status Flag	0: Transmission is pending or transmission is not requested 1: From acceptance of transmission request to completion of transmission, or error/arbitration lost	RO
TRMABT	Transmission Abort Complete Flag (3, 4)	0: Transmission has started, transmission abort failed because transmission is completed, or transmission abort is not requested 1: Transmission abort is completed	RW
— (b3)	No register bit; should be written with 0 and read as 0		—
ONESHOT	One-shot Enable Bit (5)	0: One-shot reception or one-shot transmission disabled 1: One-shot reception or one-shot transmission enabled	RW
— (b5)	No register bit; should be written with 0 and read as 0		—
RECREQ	Receive Mailbox Set Bit (4, 6, 7)	0: Not configured for reception 1: Configured for reception	RW
TRMREQ	Transmit Mailbox Set Bit (4, 6)	0: Not configured for transmission 1: Configured for transmission	RW

Notes:
1. Write to the C0MCTLj register in CAN operation mode or CAN halt mode.
2. Do not use registers C0MCTL24 to C0MCTL31 in FIFO mailbox mode.
3. Write 0 only. Writing 1 has no effect.
4. When writing 0 to bits NEWDATA, SENTDATA, MSGLOST, TRMABT, RECREQ, and TRMREQ by a program, use the MOV instruction to ensure that only the specified bit is set to 0 and the other bits are set to 1.
5. To enter one-shot receive mode, write 1 to the ONESHOT bit at the same time as setting the RECREQ bit to 1. To exit one-shot receive mode, write 0 to the ONESHOT bit after writing 0 to the RECREQ bit and confirming it has been set to 0.
   To enter one-shot transmit mode, write 1 to the ONESHOT bit at the same time as setting the TRMREQ bit to 1. To exit one-shot transmit mode, write 0 to the ONESHOT bit after the message has been transmitted or aborted.
6. Do not set both the RECREQ and TRMREQ bits to 1.
7. When setting the RECREQ bit to 0, set bits MSGLOST, NEWDATA, RECREQ to 0 simultaneously.

Figure 24.10 C0MCTLj Register

24.1.9.1 NEWDATA Bit

The NEWDATA bit is set to 1 when a new message is being stored or has been stored to the mailbox.
The timing for setting this bit to 1 is simultaneous with the INVALDATA bit.
The NEWDATA bit is set to 0 by writing 0 by a program.
This bit is not be set to 0 by writing 0 by a program while the related INVALDATA bit is 1.

24.1.9.2 SENTDATA Bit

The SENTDATA bit is set to 1 when data transmission from the corresponding mailbox is completed.
This bit is set to 0 by writing 0 by a program.
To set the SENTDATA bit to 0, first set the TRMREQ bit to 0.
Bits SENTDATA and TRMREQ cannot be set to 0 simultaneously.
To transmit a new message from the corresponding mailbox, set the SENTDATA bit to 0.

24.1.9.3 INVALDATA Bit

After the completion of a message reception, the INVALDATA bit is set to 1 while the received message is being updated into the corresponding mailbox.
This bit is set to 0 immediately after the message has been stored. If the mailbox is read while this bit is 1, the data is undefined.

24.1.9.4 TRMACTIVE Bit

The TRMACTIVE bit is set to 1 when the corresponding mailbox of the CAN module begins transmitting a message.
This bit is set to 0 when the CAN module has lost CAN bus arbitration, a CAN bus error occurs, or data transmission is completed.

24.1.9.5 MSGLOST Bit

The MSGLOST bit is set to 1 when the mailbox is overwritten or overrun by a new received message while the NEWDATA bit is 1. The MSGLOST bit is set to 1 at the end of the 6th bit of EOF.
This bit is set to 0 by writing 0 by a program.
In both overwrite and overrun modes, this bit is not set to 0 by writing 0 by a program during five cycles of fCAN (CAN system clock) following the 6th bit of EOF.

24.1.9.6 TRMABT Bit

The TRMABT bit is set to 1 in the following cases:

- Following a transmission abort request, when the transmission abort is completed before starting transmission.
- Following a transmission abort request, when the CAN module detects CAN bus arbitration lost or a CAN bus error.
- In one-shot transmission mode (RECREQ bit = 0, TRMREQ bit = 1, and ONESHOT bit = 1), when the CAN module detects CAN bus arbitration lost or a CAN bus error.

The TRMABT bit is not set to 1 when data transmission is completed. In this case, the SENTDATA bit is set to 1.
The TRMABT bit is set to 0 by writing 0 by a program.

24.1.9.7 ONESHOT Bit

The ONESHOT bit can be used in the following two ways, receive mode and transmit mode:

(1) One-shot Receive Mode

When the ONESHOT bit is set to 1 in receive mode (RECREQ bit = 1 and TRMREQ bit = 0), the mailbox receives a message only one time. The mailbox does not behave as a receive mailbox after having received a message one time. The behavior of bits NEWDATA and INVALDATA is the same as in normal reception mode. In one-shot receive mode, the MSGLOST bit is not set to 1.
To set the ONESHOT bit to 0, first write 0 to the RECREQ bit and ensure that it has been set to 0.

(2) One-shot Transmit Mode

When the ONESHOT bit is set to 1 in transmit mode (RECREQ bit = 0 and TRMREQ bit = 1), the CAN module transmits a message only one time. The CAN module does not transmit the message again if a CAN bus error or CAN bus arbitration lost occurs. When transmission is completed, the SENTDATA bit is set to 1. If transmission is not completed due to a CAN bus error or CAN bus arbitration lost, the TRMABT bit is set to 1.
Set the ONESHOT bit to 0 after the SENTDATA or TRMABT bit is set to 1.

24.1.9.8 RECREQ Bit

The RECREQ bit selects receive modes shown in Table 24.11.
When the RECREQ bit is set to 1, the corresponding mailbox is configured for reception of a data frame or a remote frame.
When this bit is set to 0, the corresponding mailbox is not configured for reception of a data frame or a remote frame.
Due to HW protection, the RECREQ bit cannot be set to 0 by writing 0 by a program during the following period:
HW protection is started
- from the acceptance filter procedure (the beginning of the CRC field)

HW protection is released
- for the mailbox that is specified to receive the incoming message, after the received data is stored into the mailbox or a CAN bus error occurs (i.e. a maximum period of HW protection is from the beginning of the CRC field to the end of the 7th bit of EOF)
- for the other mailboxes, after the acceptance filter procedure
- if no mailbox is specified to receive the message, after the acceptance filter procedure

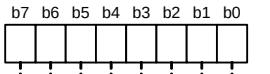
When setting the RECREQ bit to 1, do not set 1 to the TRMREQ bit.
To change the configuration of a mailbox from transmission to reception, first abort the transmission and then set bits SENTDATA and TRMABT to 0 before changing to reception.

24.1.9.9 TRMREQ Bit

The TRMREQ bit selects transmit modes shown in Table 24.11.
When this bit is set to 1, the corresponding mailbox is configured for transmission of a data frame or a remote frame.
When this bit is set to 0, the corresponding mailbox is not configured for transmission of a data frame or a remote frame.
If the TRMREQ bit is changed from 1 to 0 to cancel the corresponding transmission request, either the TRMABT or SENTDATA bit is set to 1.
When setting the TRMREQ bit to 1, do not set the RECREQ bit to 1.
To change the configuration of a mailbox from reception to transmission, first abort the reception and then set bits NEWDATA and MSGLOST to 0 before changing to transmission.

24.1.10 CAN0 Receive FIFO Control Register (C0RFCR Register)

CAN0 Receive FIFO Control Register (1)



Symbol	Address	Reset Value
C0RFCR	47F48h	1000 0000b

Bit Symbol	Bit Name	Function	RW
RFE	Receive FIFO Enable Bit (2)	0: Receive FIFO disabled 1: Receive FIFO enabled	RW
RFUST	Receive FIFO Unread Message Number Status Bit	b3 b2 b1 0 0 0 : No unread message 0 0 1 : 1 unread message 0 1 0 : 2 unread messages 0 1 1 : 3 unread messages 1 0 0 : 4 unread messages 1 0 1 : Reserved 1 1 0 : Reserved 1 1 1 : Reserved	RO
RFMLF	Receive FIFO Message Lost Flag (3)	0: No receive FIFO message lost has occurred 1: Receive FIFO message lost has occurred	RW
RFFST	Receive FIFO Full Status Bit	0: Receive FIFO is not full 1: Receive FIFO is full (4 unread messages)	RO
RFWST	Receive FIFO Buffer Warning Status Bit	0: Receive FIFO is not buffer warning 1: Receive FIFO is buffer warning (3 unread messages)	RO
RFEST	Receive FIFO Empty Status Bit	0: Unread message in receive FIFO 1: No unread message in receive FIFO	RO

Notes:
1. Write to the C0RFCR register in CAN operation mode or CAN halt mode.
2. When setting the RFE bit to 0, set the RFMLF bit to 0 as well.
3. Write 0 only. Writing 1 has no effect.

Figure 24.11 C0RFCR Register

24.1.10.1 RFE Bit

When the RFE bit is set to 1, the receive FIFO is enabled.
When this bit is set to 0, the receive FIFO is disabled for reception and becomes empty (RFEST bit = 1).
Do not set this bit to 1 in normal mailbox mode (MBM bit in the C0CTLR register = 0).
Due to HW protection, the RFE bit is not set to 0 by writing 0 by a program during the following period:
HW protection is started

- from the acceptance filter procedure (the beginning of the CRC field)

HW protection is released

- if the receive FIFO is specified to receive the incoming message, after the received data is stored into the receive FIFO or a CAN bus error occurs. (i.e. a maximum period of HW protection is from the beginning of the CRC field to the end of the 7th bit of EOF.)
- if the receive FIFO is not specified to receive the message, after the acceptance filter procedure.

24.1.10.2 RFUST Bit

The RFUST bit indicates the number of unread messages in the receive FIFO.
The value of this bit is initialized to 000b when the RFE bit is set to 0.

24.1.10.3 RFMLF Bit

The RFMLF bit is set to 1 (receive FIFO message lost has occurred) when the receive FIFO receives a new message and the receive FIFO is full. The timing for setting this bit to 1 is at the end of the 6th bit of EOF.
The RFMLF bit is set to 0 by writing 0 by a program.
In both overwrite and overrun modes, this bit cannot be set to 0 (receive FIFO message lost has not occurred) by writing 0 by a program due to HW protection during the five cycles of fCAN following the 6th bit of EOF, if the receive FIFO is full and determined to receive the message.

24.1.10.4 RFFST Bit

The RFFST bit is set to 1 (receive FIFO is full) when the number of unread messages in the receive FIFO is 4. This bit is set to 0 (receive FIFO is not full) when the number of unread messages in the receive FIFO is less than 4. This bit is set to 0 when the RFE bit is 0.

24.1.10.5 RFWST Bit

The RFWST bit is set to 1 (receive FIFO is buffer warning) when the number of unread messages in the receive FIFO is 3. This bit is set to 0 (receive FIFO is not buffer warning) when the number of unread messages in the receive FIFO is less than 3 or equal to 4. This bit is set to 0 when the RFE bit is 0.

24.1.10.6 RFEST Bit

The RFEST bit is 1 (no unread message in receive FIFO) when the number of unread messages in the receive FIFO is 0. This bit is set to 1 when the RFE bit is set to 0. The RFEST bit is set to 0 (unread message in receive FIFO) when the number of unread messages in the receive FIFO is one or more.

Figure 24.12 shows the receive FIFO mailbox operation.

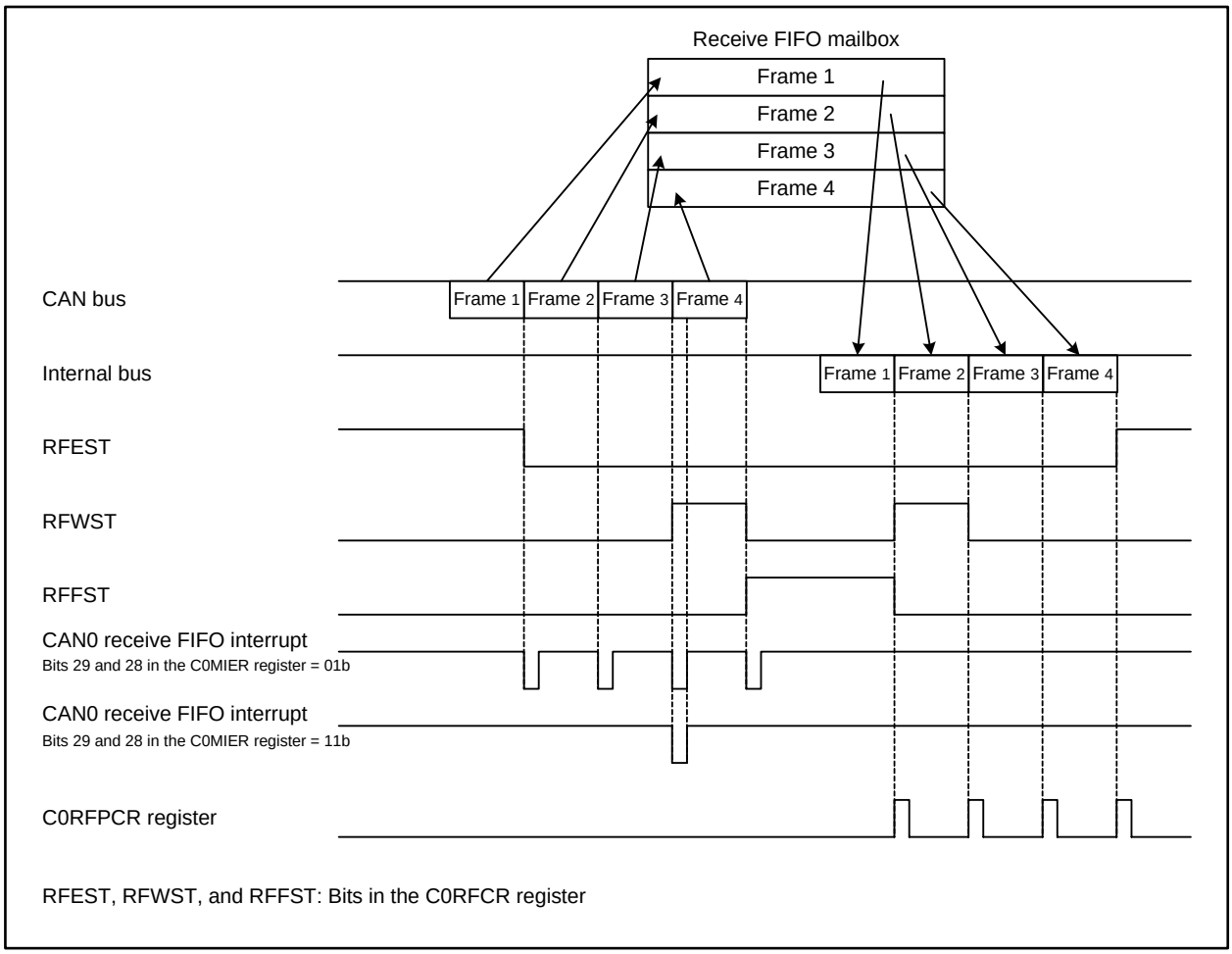
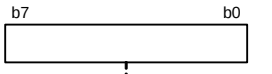


Figure 24.12 Receive FIFO Mailbox Operation (Bits 29 and 28 in C0MIER Register = 01b and 11b)

24.1.11 CAN0 Receive FIFO Pointer Control Register (C0RFPCR Register)

CAN0 Receive FIFO Pointer Control Register



Symbol	Address	Reset Value
C0RFPCR	47F49h	Undefined

Function	Setting Value	RW
The CPU-side pointer for the receive FIFO is incremented by writing FFh	FFh	WO

Figure 24.13 C0RFPCR Register

When the receive FIFO is not empty, write FFh to the C0RFPCR register by a program to increment the CPU-side pointer for the receive FIFO to the next mailbox location.
Do not write to the C0RFPCR register when the RFE bit in the C0RFCR register is 0 (receive FIFO disabled).
Both the CAN-side pointer and the CPU-side pointer are incremented when a new message is received and the RFFST bit is 1 (receive FIFO is full) in overwrite mode. When the RFMLF bit is 1 in this condition, the CPU-side pointer cannot be incremented by writing to the C0RFPCR register by a program.

24.1.12 CAN0 Transmit FIFO Control Register (C0TFCR Register)

CAN0 Transmit FIFO Control Register (1)

b7	b6	b5	b4	b3	b2	b1	b0
		0					

Symbol	Address	Reset Value
C0TFCR	47F4Ah	1000 0000b

Bit Symbol	Bit Name	Function	RW
TFE	Transmit FIFO Enable Bit	0: Transmit FIFO disabled 1: Transmit FIFO enabled	RW
TFUST	Transmit FIFO Unsent Message Number Status Bit	b3 b2 b1 0 0 0 : No unsent message 0 0 1 : 1 unsent message 0 1 0 : 2 unsent messages 0 1 1 : 3 unsent messages 1 0 0 : 4 unsent messages 1 0 1 : Reserved 1 1 0 : Reserved 1 1 1 : Reserved	RO
— (b4)	No register bit; should be written with 0 and read as 0		—
— (b5)	Reserved	Should be written with 0 and read as undefined value	RO
TFFST	Transmit FIFO Full Status Bit	0: Transmit FIFO is not full 1: Transmit FIFO is full (4 unsent messages)	RO
TFEST	Transmit FIFO Empty Status Bit	0: Unsent message in transmit FIFO 1: No unsent message in transmit FIFO	RO

Note:
1. Write to the C0TFCR register in CAN operation mode or CAN halt mode.

Figure 24.14 C0TFCR Register

24.1.12.1 TFE Bit

When the TFE bit is set to 1, the transmit FIFO is enabled.
When this bit is set to 0, the transmit FIFO becomes empty (TFEST bit = 1) and then unsent messages from the transmit FIFO are lost as described below:

- If a message from the transmit FIFO is not scheduled for the next transmission or during transmission.
- Following the completion of transmission, a CAN bus error, CAN bus arbitration lost, or entry to CAN halt mode if a message from the transmit FIFO is scheduled for the next transmission or already during transmission.

Before setting the TFE bit to set to 1 again, ensure that the TFEST bit has been set to 1.
After setting the TFE bit to 1, write transmit data into the C0MB24 register.
Do not set this bit to 1 in normal mailbox mode (MBM bit in the C0CTLR register = 0).

24.1.12.2 TFUST Bit

The TFUST bit indicates the number of unsent messages in the transmit FIFO.
After the TFE bit is set to 0, the value of the TFUST bit is initialized to 000b when transmission abort or transmission is completed.

24.1.12.3 TFFST Bit

The TFFST bit is set to 1 (transmit FIFO is full) when the number of unsent messages in the transmit FIFO is 4. This bit is set to 0 (transmit FIFO is not full) when the number of unsent messages in the transmit FIFO is less than 4. This bit is set to 0 when transmission from the transmit FIFO has been aborted.

24.1.12.4 TFEST Bit

The TFEST bit is set to 1 (no message in transmit FIFO) when the number of unsent messages in the transmit FIFO is 0. This bit is set to 1 when transmission from the transmit FIFO has been aborted.
The TFEST bit is set to 0 (message in transmit FIFO) when the number of unsent messages in the transmit FIFO is not 0.

Figure 24.15 shows the transmit FIFO mailbox operation.

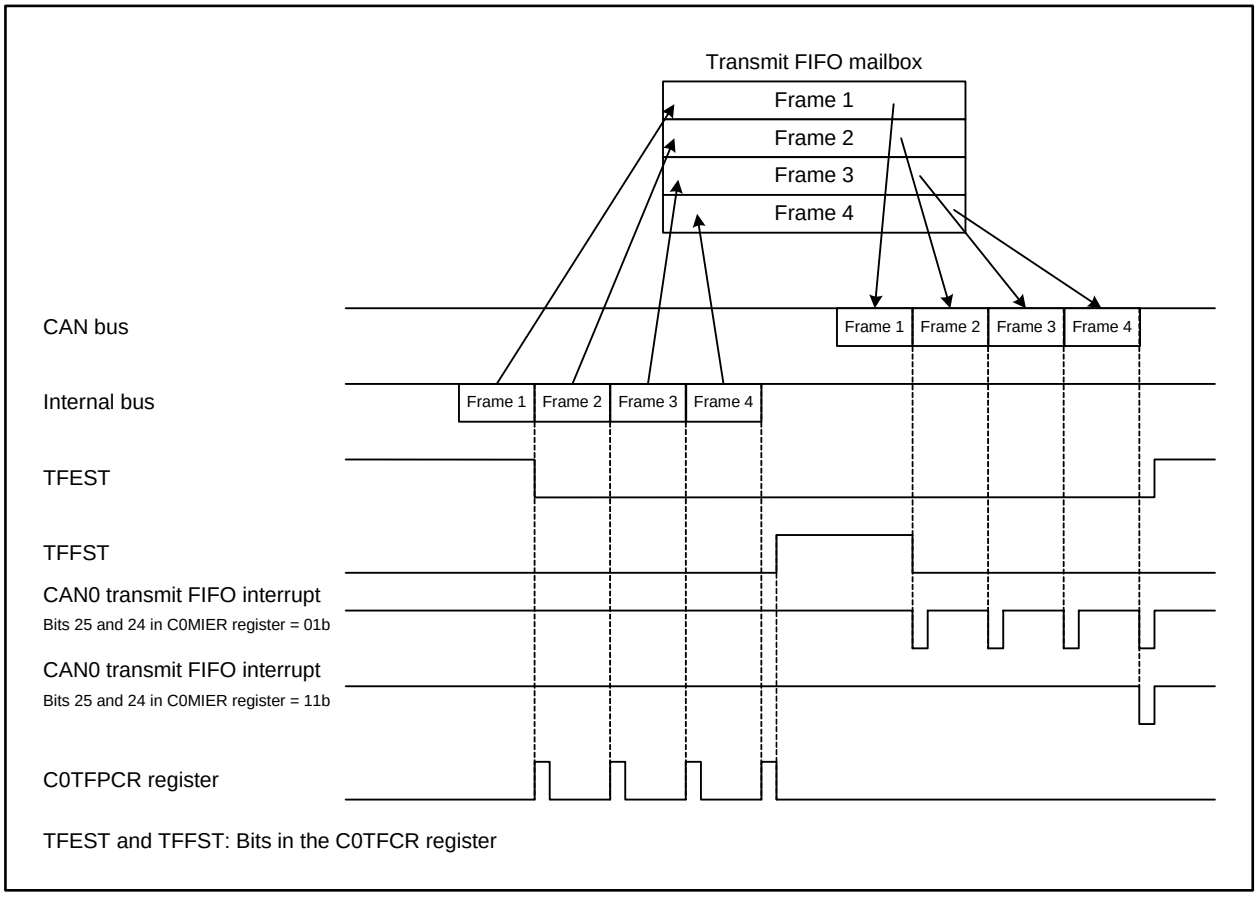
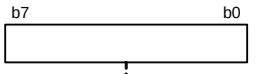


Figure 24.15 Transmit FIFO Mailbox Operation (Bits 25 and 24 in C0MIER Register = 01b and 11b)

24.1.13 CAN0 Transmit FIFO Pointer Control Register (C0TFPCR Register)

CAN0 Transmit FIFO Pointer Control Register



Symbol	Address	Reset Value
C0TFPCR	47F4Bh	Undefined

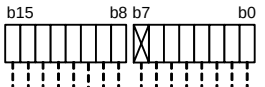
Function	Setting Value	RW
The CPU-side pointer for the transmit FIFO is incremented by writing FFh	FFh	WO

Figure 24.16 C0TFPCR Register

When the transmit FIFO is not full, write FFh to the C0TFPCR register by a program to increment the CPU-side pointer for the transmit FIFO to the next mailbox location.
Do not write to the C0TFPCR register when the TFE bit in the C0TFCR register is 0 (transmit FIFO disabled).

24.1.14 CAN0 Status Register (C0STR Register)

CAN0 Status Register



Symbol	Address	Reset Value
C0STR	47F43h-47F42h	0000 0000 0000 0101b

Bit Symbol	Bit Name	Function	RW
RSTST	CAN Reset Status Flag	0: Not in CAN reset mode 1: In CAN reset mode	RO
HLTST	CAN Halt Status Flag	0: Not in CAN halt mode 1: In CAN halt mode	RO
SLPST	CAN Sleep Status Flag	0: Not in CAN sleep mode 1: In CAN sleep mode	RO
EPST	Error-passive Status Flag	0: Not in error-passive state 1: In error-passive state	RO
BOST	Bus-off Status Flag	0: Not in bus-off state 1: In bus-off state	RO
TRMST	Transmit Status Flag (transmitter)	0: Bus idle or reception in progress 1: Transmission in progress or in bus-off state	RO
RECST	Receive Status Flag (receiver)	0: Bus idle or transmission in progress 1: Reception in progress	RO
— (b7)	No register bit; should be written with 0 and read as 0		—
NDST	NEWDATA Status Flag	0: No mailbox with NEWDATA bit = 1 1: Mailbox(es) with NEWDATA bit = 1	RO
SDST	SENTDATA Status Flag	0: No mailbox with SENTDATA bit = 1 1: Mailbox(es) with SENTDATA bit = 1	RO
RFST	Receive FIFO Status Flag	0: No message in receive FIFO 1: Message in receive FIFO	RO
TFST	Transmit FIFO Status Flag	0: Transmit FIFO is full 1: Transmit FIFO is not full	RO
NMLST	Normal Mailbox Message Lost Status Flag	0: No mailbox with MSGLOST bit = 1 1: Mailbox(es) with MSGLOST bit = 1	RO
FMLST	FIFO Mailbox Message Lost Status Flag	0: RFMLF bit = 0 1: RFMLF bit = 1	RO
TABST	Transmission Abort Status Flag	0: No mailbox with TRMABT bit = 1 1: Mailbox(es) with TRMABT bit = 1	RO
EST	Error Status Flag	0: No error occurred 1: Error occurred	RO

Figure 24.17 C0STR Register

24.1.14.1 RSTST Bit

The RSTST bit is set to 1 when the CAN module is in CAN reset mode.
This bit is set to 0 when the CAN module is not in CAN reset mode.
Even when the state is changed from CAN reset mode to CAN sleep mode, the RSTST bit remains 1.

24.1.14.2 HLTST Bit

The HLTST bit is set to 1 when the CAN module is in CAN halt mode.
This bit is set to 0 when the CAN module is not in CAN halt mode.
Even when the state is changed from CAN halt mode to CAN sleep mode, the HLTST bit remains 1.

24.1.14.3 SLPST Bit

The SLPST bit is set to 1 when the CAN module is in CAN sleep mode.
This bit is set to 0 when the CAN module is not in CAN sleep mode.

24.1.14.4 EPST Bit

The EPST bit is set to 1 when the value of the C0TECR or C0RECR register exceeds 127 and the CAN module is in error-passive state ($128 \leq TEC < 256$ or $128 \leq REC < 256$). This bit is set to 0 when the CAN module is not in error-passive state.
TEC indicates the value of the transmit error counter (C0TECR register) and REC indicates the value of the receive error counter (C0RECR register).

24.1.14.5 BOST Bit

The BOST bit is set to 1 when the value of the C0TECR register exceeds 255 and the CAN module is in bus-off state ($TEC \geq 256$). This bit is set to 0 when the CAN module is not in bus-off state.

24.1.14.6 TRMST Bit

The TRMST bit is set to 1 when the CAN module performs as a transmitter node or is in bus-off state.
This bit is set to 0 when the CAN module performs as a receiver node or is in bus-idle state.

24.1.14.7 RECST Bit

The RECST bit is set to 1 when the CAN module performs as a receiver node.
This bit is set to 0 when the CAN module performs as a transmitter node or is in bus-idle state.

24.1.14.8 NDST Bit

The NDST bit is set to 1 when at least one NEWDATA bit in the C0MCTLj register (j = 0 to 31) is 1 regardless of the value of the C0MIER register.
The NDST bit is set to 0 when all NEWDATA bits are 0.

24.1.14.9 SDST Bit

The SDST bit is set to 1 when at least one SENTDATA bit in the C0MCTLj register (j = 0 to 31) is 1 regardless of the value of the C0MIER register.
The SDST bit is set to 0 when all SENTDATA bits are 0.

24.1.14.10 RFST Bit

The RFST bit is set to 1 when the receive FIFO is not empty.
This bit is set to 0 when the receive FIFO is empty.
This bit is set to 0 when normal mailbox mode is selected.

24.1.14.11 TFST Bit

The TFST bit is set to 1 when the transmit FIFO is not full.
This bit is set to 0 when the transmit FIFO is full.
This bit is set to 0 when normal mailbox mode is selected.

24.1.14.12 NMLST Bit

The NMLST bit is set to 1 when at least one MSGLOST bit in the C0MCTLj register is 1 regardless of the value of the C0MIER register.
The NMLST bit is set to 0 when all MSGLOST bits are 0.

24.1.14.13 FMLST Bit

The FMLST bit is set to 1 when the RFMLF bit in the C0RFCR register is 1 regardless of the value of the C0MIER register.
The FMLST bit is set to 0 when the RFMLF bit is 0.

24.1.14.14 TABST Bit

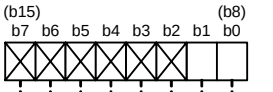
The TABST bit is set to 1 when at least one TRMABT bit in the C0MCTLj register is 1 regardless of the value of the C0MIER register.
The TABST bit is set to 0 when all TRMABT bits are 0.

24.1.14.15 EST Bit

The EST bit is 1 when at least one error is detected by the C0EIFR register regardless of the value of the C0EIER register.
This bit is set to 0 when no error is detected by the C0EIFR register.

24.1.15 CAN0 Mailbox Search Mode Register (C0MSMR Register)

CAN0 Mailbox Search Mode Register (1)



Symbol	Address	Reset Value
C0MSMR	47F53h	0000 0000b

Bit Symbol	Bit Name	Function	RW
MBSM	Mailbox Search Mode Select Bit	b1 b0 0 0 : Receive mailbox search mode 0 1 : Transmit mailbox search mode 1 0 : Message lost search mode 1 1 : Channel search mode	RW
— (b7-b2)	No register bits; should be written with 0 and read as 0		—

Note:
1. Write to the C0MSMR register in CAN operation mode or CAN halt mode.

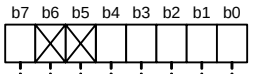
Figure 24.18 C0MSMR Register

24.1.15.1 MBSM Bit

The MBSM bit selects the search mode for the mailbox search function.
When this bit is 00b, receive mailbox search mode is selected. In this mode, the search targets are the NEWDATA bit in the C0MCTLj register (j = 0 to 31) for the normal mailbox and the RFEST bit in the C0RFCR register.
When the MBSM bit is 01b, transmit mailbox search mode is selected. In this mode, the search target is the SENTDATA bit in the C0MCTLj register.
When the MBSM bit is 10b, message lost search mode is selected. In this mode, the search targets are the MSGLOST bit in the C0MCTLj register for the normal mailbox and the RFMLF bit in the C0RFCR register.
When the MBSM bit is 11b, channel search mode is selected. In this mode, the search target is the C0CSSR register. Refer to 24.1.17 “CAN0 Channel Search Support Register (C0CSSR Register)”.

24.1.16 CAN0 Mailbox Search Status Register (C0MSSR Register)

CAN0 Mailbox Search Status Register



Symbol	Address	Reset Value
C0MSSR	47F52h	1000 0000b

Bit Symbol	Bit Name	Function	RW
MBNST	Search Result Mailbox Number Status Bit	Output of search result in each search mode Output number: 0 to 31	RO
— (b6-b5)	No register bits; should be written with 0 and read as 0		—
SEST	Search Result Status Bit	0: Search result found 1: No search result	RO

Figure 24.19 C0MSSR Register

24.1.16.1 MBNST Bit

The MBNST bit outputs the smallest mailbox number that is searched in each mode of the C0MSMR register.
In receive mailbox, transmit mailbox, and message lost search modes, the value of the mailbox i.e.,the search result to be output, is updated as described below:

- When the NEWDATA, SENTDATA, or MSGLOST bit for the output mailbox is set to 0.
- When the NEWDATA, SENTDATA, or MSGLOST bit for a higher-priority mailbox is set to 1.

In receive mailbox search and message lost search modes, the receive FIFO (mailbox [28]) is output when the receive FIFO is not empty and there are no unread received messages or no lost messages in any of the normal mailboxes (mailboxes [0] to [23]).
In transmit mailbox search mode, the transmit FIFO (mailbox [24]) is not output.
Table 24.7 lists the behavior of MBNST bit in FIFO mailbox mode.

Table 24.7 Behavior of MBNST Bit in FIFO Mailbox Mode

MBSM Bit	Mailbox [24] (Transmit FIFO)	Mailbox [28] (Receive FIFO)
00b	Mailbox [24] is not output	Mailbox [28] is output when no NEWDATA bit for the normal mailbox is set to 1 and the receive FIFO is not empty
01b		Mailbox [28] is not output
10b		Mailbox [28] is output when no MSGLOST bit for the normal mailbox is set to 1 and the RFMLF bit is set to 1 in the receive FIFO
11b		Mailbox [28] is not output

In channel search mode, the MBNST bit outputs the corresponding channel number. After the C0MSSR register is read by a program, the next target channel number is output.

24.1.16.2 SEST Bit

The SEST bit is set to 1 when no corresponding mailbox is found after searching all mailboxes.
For example, in transmit mailbox search mode, the SEST bit is set to 1 when no SENTDATA bit for mailboxes is 1. The SEST bit is set to 0 when at lease one SENTDATA bit is 1.
When the SEST bit is 1, the value of the MBNST bit is undefined.

24.1.17 CAN0 Channel Search Support Register (C0CSSR Register)

CAN0 Channel Search Support Register (1, 2)

b7 b0

Symbol	Address	Reset Value
C0CSSR	47F51h	Undefined

Function	Setting Value	RW
When the value of the channel search is input, the channel number is output to the CiMSSR register	Channel value	RW

Notes:
1. Write to the C0CSSR register only when the MBSM bit in the C0MSMR register is 11b (channel search mode).
2. Write to the C0CSSR register in CAN operation mode and CAN halt mode.

Figure 24.20 C0CSSR Register

The bits in the C0CSSR register, which are set to 1, are encoded by an 8-to-3 priority encoder (the lower bit position, the higher priority) and output to the MBNST bits in the C0MSSR register.
The value of the C0MSSR register is updated whenever the C0MSSR register is read.
Figure 24.21 shows the write and read of registers C0CSSR and C0MSSR.

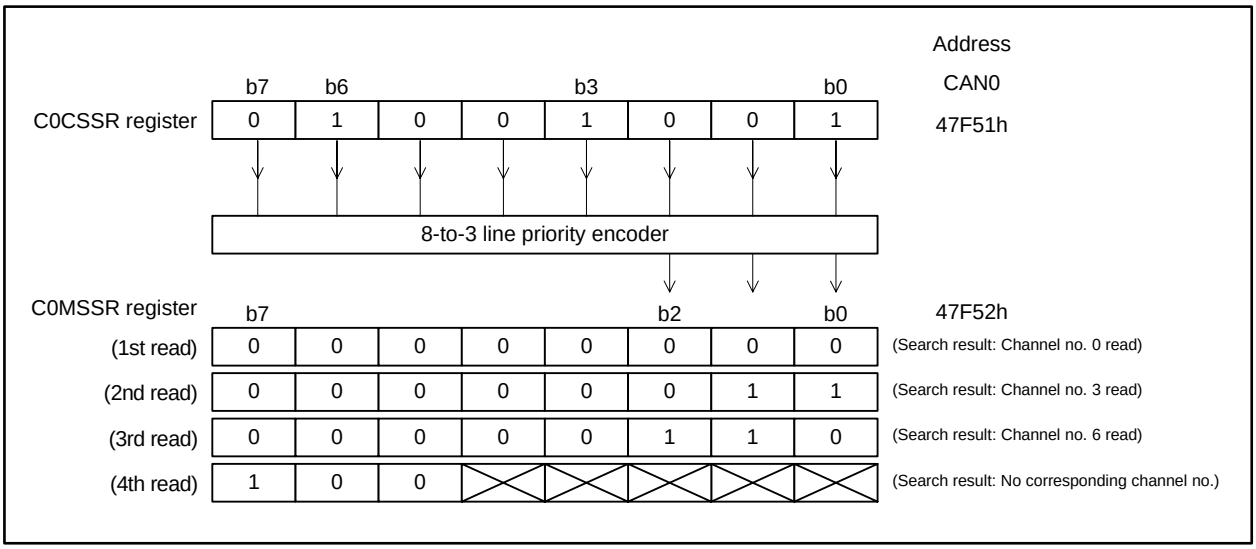
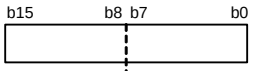

Address
b7 b6 b3 b0
CAN0
C0CSSR register
0 1 0 0 1 0 0 1
47F51h
8-to-3 line priority encoder
C0MSSR register
b7 b2 b0
47F52h
(1st read) 0 0 0 0 0 0 0 0 (Search result: Channel no. 0 read)
(2nd read) 0 0 0 0 0 0 1 1 (Search result: Channel no. 3 read)
(3rd read) 0 0 0 0 0 1 1 0 (Search result: Channel no. 6 read)
(4th read) 1 0 0 (Search result: No corresponding channel no.)

Figure 24.21 Write and Read of Registers C0CSSR and C0MSSR

The value of the C0CSSR register is also updated whenever the C0MSSR register is read. When the C0CSSR register is read, the value before the 8-to-3 priority encoder conversion is read.

24.1.18 CAN0 Acceptance Filter Support Register (C0AFSR Register)

CAN0 Acceptance Filter Support Register (1)



Symbol	Address	Reset Value
C0AFSR	47F57h-47F56h	Undefined

Function	Setting Value	RW
After the standard ID of a received message is written, the value converted for data table search can be read	Standard ID/ converted value	RW

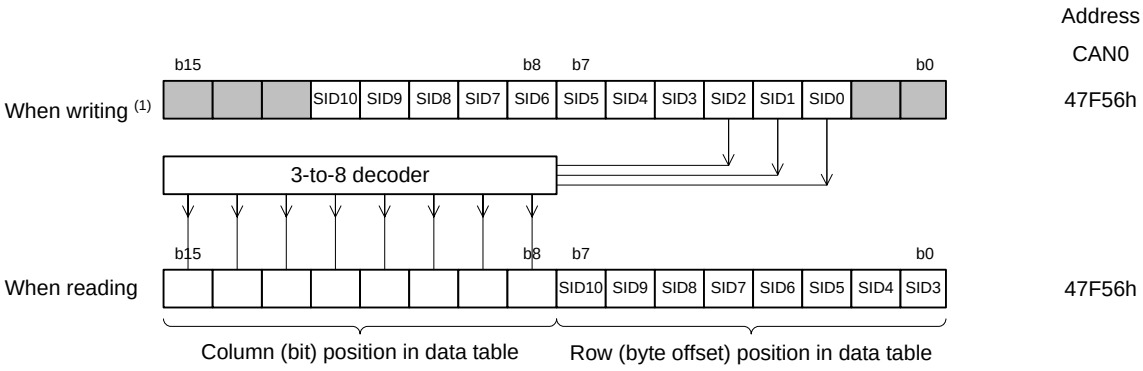
Note:
1. Write to the C0AFSR register in CAN operation mode or CAN halt mode.

Figure 24.22 C0AFSR Register

The acceptance filter support unit (ASU) can be used for data table (8 bits × 256) search. In the data table, all standard IDs created by the user are set to be valid/invalid in bit units. When the C0AFSR register is written with the 16-bit unit data including the SID bit in the C0MBj register (j = 0 to 31), in which a received ID is stored, a decoded row (byte offset) position and column (bit) position for data table search can be read. The ASU can be used for standard (11-bit) IDs only.
The ASU is enabled in the following cases:

- When the ID to receive cannot be masked by the acceptance filter.
  Example) IDs to receive: 078h, 087h, 111h
- When there are too many IDs to receive and software filtering time is expected to be shortened.

Figure 24.23 shows the write and read of C0AFSR register.

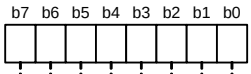


Note:
1. Write the same value as the 16-bit unit data including the SID bit in the C0MBj (j = 0 to 31) register.

Figure 24.23 Write and Read of C0AFSR Register

24.1.19 CAN0 Error Interrupt Enable Register (C0EIER Register)

CAN0 Error Interrupt Enable Register (1)



Symbol	Address	Reset Value
C0EIER	47F4Ch	00h

Bit Symbol	Bit Name	Function	RW
BEIE	Bus Error Interrupt Enable Bit	0: Bus error interrupt disabled 1: Bus error interrupt enabled	RW
EWIE	Error Warning Interrupt Enable Bit	0: Error warning interrupt disabled 1: Error warning interrupt enabled	RW
EPIE	Error Passive Interrupt Enable Bit	0: Error passive interrupt disabled 1: Error passive interrupt enabled	RW
BOEIE	Bus-off Entry Interrupt Enable Bit	0: Bus-off entry interrupt disabled 1: Bus-off entry interrupt enabled	RW
BORIE	Bus-off Recovery Interrupt Enable Bit	0: Bus-off recovery interrupt disabled 1: Bus-off recovery interrupt enabled	RW
ORIE	Receive Overrun Interrupt Enable Bit	0: Receive overrun interrupt disabled 1: Receive overrun interrupt enabled	RW
OLIE	Overload Frame Transmit Interrupt Enable Bit	0: Overload frame transmit interrupt disabled 1: Overload frame transmit interrupt enabled	RW
BLIE	Bus Lock Interrupt Enable Bit	0: Bus lock interrupt disabled 1: Bus lock interrupt enabled	RW

Note:
1. Write to the C0EIER register in CAN reset mode.

Figure 24.24 C0EIER Register

The C0EIER register is used to set the error interrupt enabled/disabled individually for each error interrupt source in the C0EIFR register.

24.1.19.1 BEIE Bit

When the BEIE bit is 0, no error interrupt request is generated even if the BEIF bit in the C0EIFR register is set to 1.
When the BEIE bit is 1, an error interrupt request is generated if the BEIF bit is set to 1.

24.1.19.2 EWIE Bit

When the EWIE bit is 0, no error interrupt request is generated even if the EWIF bit in the C0EIFR register is set to 1.
When the EWIE bit is 1, an error interrupt request is generated if the EWIF bit is set to 1.

24.1.19.3 EPIE Bit

When the EPIE bit is 0, no error interrupt request is generated even if the EPIF bit in the C0EIFR register is set to 1.
When the EPIE bit is 1, an error interrupt request is generated if the EPIF bit is set to 1.

24.1.19.4 BOEIE Bit

When the BOEIE bit is 0, no error interrupt request is generated even if the BOEIF bit in the C0EIFR register is set to 1.
When the BOEIE bit is 1, an error interrupt request is generated if the BOEIF bit is set to 1.

24.1.19.5 BORIE Bit

When the BORIE bit is 0, an error interrupt request is not generated even if the BORIF bit in the C0EIFR register is set to 1.
When the BORIE bit is 1, an error interrupt request is generated if the BORIF bit is set to 1.

24.1.19.6 ORIE Bit

When the ORIE bit is 0, no error interrupt request is generated even if the ORIF bit in the C0EIFR register is set to 1.
When the ORIE bit is 1, an error interrupt request is generated if the ORIF bit is set to 1.

24.1.19.7 OLIE Bit

When the OLIE bit is 0, no error interrupt request is generated even if the OLIF bit in the C0EIFR register is set to 1.
When the OLIE bit is 1, an error interrupt request is generated if the OLIF bit is set to 1.

24.1.19.8 BLIE Bit

When the BLIE bit is 0, no error interrupt request is generated even if the BLIF bit in the C0EIFR register is set to 1.
When the BLIE bit is 1, an error interrupt request is generated if the BLIF bit is set to 1.

24.1.20 CAN0 Error Interrupt Factor Judge Register (C0EIFR Register)

CAN0 Error Interrupt Factor Judge Register (1)

b7	b6	b5	b4	b3	b2	b1	b0

Symbol	Address	Reset Value
C0EIFR	47F4Dh	00h

Bit Symbol	Bit Name	Function	RW
BEIF	Bus Error Detect Flag	0: No bus error detected 1: Bus error detected	RW
EWIF	Error Warning Detect Flag	0: No error warning detected 1: Error warning detected	RW
EPIF	Error Passive Detect Flag	0: No error passive detected 1: Error passive detected	RW
BOEIF	Bus-off Entry Detect Flag	0: No bus-off entry detected 1: Bus-off entry detected	RW
BORIF	Bus-off Recovery Detect Flag	0: No bus-off recovery detected 1: Bus-off recovery detected	RW
ORIF	Receive Overrun Detect Flag	0: No receive overrun detected 1: Receive overrun detected	RW
OLIF	Overload Frame Transmission Detect Flag	0: No overload frame transmission detected 1: Overload frame transmission detected	RW
BLIF	Bus Lock Detect Flag	0: No bus lock detected 1: Bus lock detected	RW

Note:
1. When writing 0 to these bits by a program, use the MOV instruction to ensure that only the specified bit is set to 0 and the other bits are set to 1. Writing 1 has no effect to these bit values.

Figure 24.25 C0EIFR Register

If an event corresponding to each bit occurs, the corresponding bit in the C0EIFR register is set to 1 regardless of the setting of the C0EIER register.
To set each bit to 0, write 0 by a program. If the set timing occurs simultaneously with the clear timing by the program, the bit becomes 1.

24.1.20.1 BEIF Bit

The BEIF bit is set to 1 when a bus error is detected.

24.1.20.2 EWIF Bit

The EWIF bit is set to 1 when the value of the receive error counter (REC) or transmit error counter (TEC) exceeds 95.
This bit is set to 1 only when the REC or TEC initially exceeds 95. Thus, if 0 is written to the EWIF bit by a program while the REC or TEC remains greater than 95, this bit is not set to 1 until the REC and the TEC go below 95 and then exceed 95 again.

24.1.20.3 EPIF Bit

The EPIF bit is set to 1 when the CAN error state becomes error-passive (the REC or TEC value exceeds 127).
This bit is set to 1 only when the REC or TEC initially exceeds 127. Thus, if 0 is written to the EPIF bit by a program while the REC or TEC remains greater than 127, this bit is not set to 1 until the REC and the TEC go below 127 and then exceed 127 again.

24.1.20.4 BOEIF Bit

The BOEIF bit is set to 1 when the CAN error state becomes bus-off (the TEC value exceeds 255).
This bit is also set to 1 when the BOM bit in the C0CTLR register is 01b (entry to CAN halt mode automatically at bus-off entry) and the CAN module becomes the bus-off state.

24.1.20.5 BORIF Bit

The BORIF bit is set to 1 when the CAN module recovers from the bus-off state normally by detecting 11 consecutive bits 128 times in the following conditions:
(1) When the BOM bit in the C0CTLR register is 00b
(2) When the BOM bit is 10b
(3) When the BOM bit is 11b

The BORIF bit is not set to 1 if the CAN module recovers from the bus-off state in the following conditions:
(1) When the CANM bit in the C0CTLR register is set to 01b (CAN reset mode)
(2) When the RBOC bit in the C0CTLR register is set to 1 (forcible return from bus-off)
(3) When the BOM bit is 01b
(4) When the BOM bit is 11b and the CANM bit is set to 10b (CAN halt mode) before normal recovery occurs

Table 24.8 lists the behavior of bits BOEIF and BORIF according to BOM bit setting value.

Table 24.8 Behavior of Bits BOEIF and BORIF according to BOM Bit Setting Value

BOM Bit	BOEIF Bit	BORIF Bit
00b	Set to 1 on entry to the bus-off state.	Set to 1 on exit from the bus-off state.
01b		Do not set to 1.
10b		Set to 1 on exit from the bus-off state.
11b		Set to 1 if normal bus-off recovery occurs before the CANM bit is set to 10b (CAN halt mode).

24.1.20.6 ORIF Bit

The ORIF bit is set to 1 when a receive overrun occurs.
This bit is not to set to 1 in overwrite mode. In overwrite mode, a reception complete interrupt request is generated if an overwrite condition occurs and this bit is not set to 1.
In normal mailbox mode, if an overrun occurs in any of mailboxes [0] to [31] in overrun mode, this bit is set to 1.
In FIFO mailbox mode, if an overrun occurs in any of mailboxes [0] to [23] or the receive FIFO in overrun mode, this bit is set to 1.

24.1.20.7 OLIF Bit

The OLIF bit is set to 1 if the transmitting condition of an overload frame is detected when the CAN module performs transmission or reception.

24.1.20.8 BLIF Bit

The BLIF bit is set to 1 if 32 consecutive dominant bits are detected on the CAN bus while the CAN module is in CAN operation mode.

After the BLIF bit is set to 1, 32 consecutive dominant bits are detected again under either of the following conditions:

- After this bit is set to 0 from 1, recessive bits are detected.
- After this bit is set to 0 from 1, the CAN module enters CAN reset mode or CAN halt mode and then enters CAN operation mode again.

24.1.21 CAN0 Receive Error Count Register (C0RECR Register)

CAN0 Receive Error Count Register

b7 b0

Symbol	Address	Reset Value
C0RECR	47F4Eh	00h

Function	Counter Value	RW
Receive error count function The C0RECR register increments or decrements the counter value according to the error status of the CAN module during reception	00h to FFh (1)	RO

Note:
1. The value in bus-off state is undefined.

Figure 24.26 C0RECR Register

The C0RECR register indicates the value of the receive error counter.
Refer to the CAN Specifications (ISO11898-1) about the increment/decrement conditions of the receive error counter.

24.1.22 CAN0 Transmit Error Count Register (C0TECR Register)

CAN0 Transmit Error Count Register

b7 b0

Symbol	Address	Reset Value
C0TECR	47F4Fh	00h

Function	Counter Value	RW
Transmit error count function The C0TECR register increments or decrements the counter value according to the error status of the CAN module during transmission	00h to FFh (1)	RO

Note:
1. The value in bus-off state is undefined.

Figure 24.27 C0TECR Register

The C0TECR register indicates the value of the TEC error counter.
Refer to the CAN Specifications (ISO11898-1) about the increment/decrement conditions of the transmit error counter.

24.1.23 CAN0 Error Code Store Register (C0ECSR Register)

CAN0 Error Code Store Register

b7	b6	b5	b4	b3	b2	b1	b0

Symbol	Address	Reset Value
C0ECSR	47F50h	00h

Bit Symbol	Bit Name	Function	RW
SEF	Stuff Error Flag (1, 2)	0: No stuff error detected 1: Stuff error detected	RW
FEF	Form Error Flag (1, 2)	0: No form error detected 1: Form error detected	RW
AEF	ACK Error Flag (1, 2)	0: No ACK error detected 1: ACK error detected	RW
CEF	CRC Error Flag (1, 2)	0: No CRC error detected 1: CRC error detected	RW
BE1F	Bit Error (recessive) Flag (1, 2)	0: No bit error detected 1: Bit error (recessive) detected	RW
BE0F	Bit Error (dominant) Flag (1, 2)	0: No bit error detected 1: Bit error (dominant) detected	RW
ADEF	ACK Delimiter Error Bit (1, 2)	0: No ACK delimiter error detected 1: ACK delimiter error detected	RW
EDPM	Error Display Mode Select Bit (3)	0: Output of first detected error code (4) 1: Output of accumulated error code	RW

Notes:
1. Writing 1 has no effect to these bit values.
2. When writing 0 to bits SEF, FEF, AEF, CEF, BE1F, BE0F, and ADEF by a program, use the MOV instruction to ensure that only the specified bit is set to 0 and the other bits are set to 1.
3. Write to the EDPM bit in CAN reset mode or CAN halt mode.
4. If more than one error condition is detected simultaneously, all corresponding bits are set to 1.

Figure 24.28 C0ECSR Register

The C0ECSR register can be used to monitor whether an error has occurred on the CAN bus. Refer to the CAN Specifications (ISO11898-1) to check the generation conditions of each error.
To set each bit except the EDPM bit to 0, write 0 by a program. If the timing at which each bit is set to 1 and the timing at which is written by a program are the same, the relevant bit is set to 1.

24.1.23.1 SEF Bit

The SEF bit is set to 1 when a stuff error is detected.

24.1.23.2 FEF Bit

The FEF bit is set to 1 when a form error is detected.

24.1.23.3 AEF Bit

The AEF bit is set to 1 when an ACK error is detected.

24.1.23.4 CEF Bit

The CEF bit is set to 1 when a CRC error is detected.

24.1.23.5 BE1F Bit

The BE1F bit is set to 1 when a recessive bit error is detected.

24.1.23.6 BE0F Bit

The BE0F bit is set to 1 when a dominant bit error is detected.

24.1.23.7 ADEF Bit

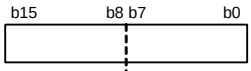
The ADEF bit is set to 1 when a form error is detected with the ACK delimiter during transmission.

24.1.23.8 EDPM Bit

The EDPM bit selects the output mode of the C0ECSR register.
When this bit is set to 0, the C0ECSR register outputs the first error code.
When this bit is set to 1, the C0ECSR register outputs the accumulated error code.

24.1.24 CAN0 Time Stamp Register (C0TSR Register)

CAN0 Time Stamp Register (1)



Symbol	Address	Reset Value
C0TSR	47F55h-47F54h	0000h

Function	Counter Value	RW
Free-running counter value for the time stamp function	0000h to FFFFh	RO

Note:
1. Read the C0TSR register in 16-bit units.

Figure 24.29 C0TSR Register

When the C0TSR register is read, the value of the time stamp counter (16-bit free-running counter) at that moment is read.
The value of the time stamp counter reference clock is a multiple of 1 bit time, as configured by the TSPS bit in the C0CTLR register.
The time stamp counter stops in CAN sleep mode and CAN halt mode, and is initialized in CAN reset mode.
The time stamp counter value is stored to TSL and TSH in the C0MBj register (j = 0 to 31) when a received message is stored in a receive mailbox.

24.1.25 CAN0 Test Control Register (C0TCR Register)

CAN0 Test Control Register (1)

b7	b6	b5	b4	b3	b2	b1	b0
0	0	0	0	0			

Symbol	Address	Reset Value
C0TCR	47F58h	00h

Bit Symbol	Bit Name	Function	RW
TSTE	CAN Test Mode Enable Bit	0: CAN test mode disabled 1: CAN test mode enabled	RW
TSTM	CAN Test Mode Select Bit	b2 b1 0 0 : Other than CAN test mode 0 1 : Listen only mode 1 0 : Self test mode 0 (external loop back) 1 1 : Self test mode 1 (internal loop back)	RW
— (b7-b3)	Reserved	Should be written with 0	RW

Note:
1. Write to the C0TCR register only in CAN halt mode.

Figure 24.30 C0TCR Register

24.1.25.1 TSTE Bit

When the TSTE bit is set to 0, CAN test mode is disabled.
When this bit is set to 1, CAN test mode is enabled.

24.1.25.2 TSTM Bit

The TSTM bit selects the CAN test mode.
The details of each CAN test mode is described below.

24.1.25.3 Listen Only Mode

The ISO 11898-1 recommends an optional bus monitoring mode. In listen only mode, the CAN node is able to receive valid data frames and valid remote frames. It sends only recessive bits on the CAN bus and the protocol controller is not required to send the ACK bit, overload flag, or active error flag.
Listen only mode can be used for baud rate detection.
Do not request transmission from any mailboxes in this mode.
Figure 24.31 shows the connection when listen only mode is selected.

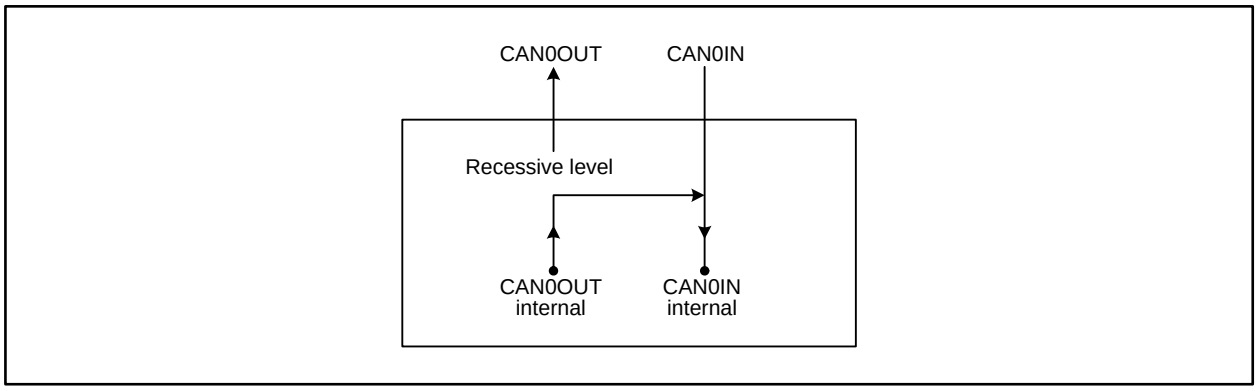


Figure 24.31 Connection when Listen Only Mode is Selected

24.1.25.4 Self Test Mode 0 (External Loop Back)

Self test mode 0 is provided for CAN transceiver tests.
In this mode, the protocol controller treats its own transmitted messages as messages received via the CAN transceiver and stores them into the receive mailbox. To be independent from external stimulation, the protocol controller generates the ACK bit.
Connect the CAN0OUT/CAN0IN pins to the transceiver.
Figure 24.32 shows the connection when self test mode 0 is selected.

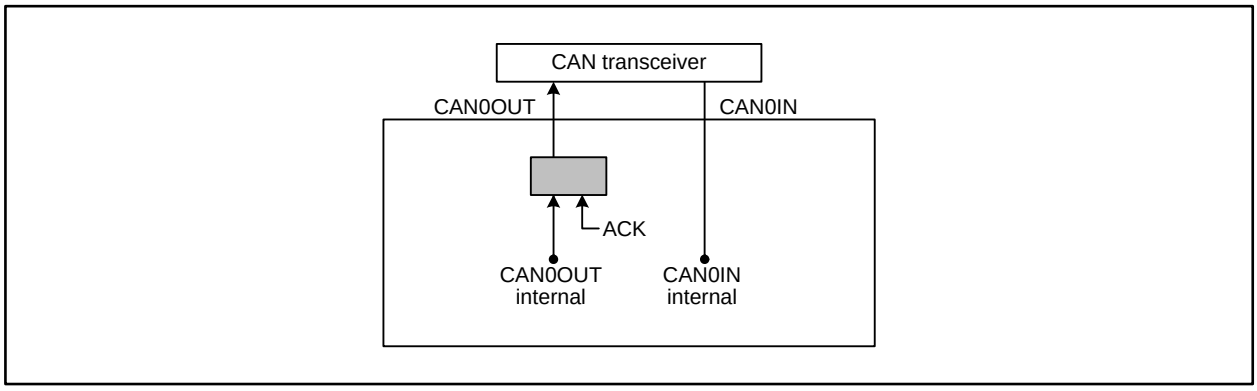


Figure 24.32 Connection when Self Test Mode 0 is Selected

24.1.25.5 Self Test Mode 1 (Internal Loop Back)

Self test mode 1 is provided for self test functions.

In this mode, the protocol controller treats its transmitted messages as received messages and stores them into the receive mailbox. To be independent from external stimulation, the protocol controller generates the ACK bit.

In self test mode 1, the protocol controller performs an internal feedback from the internal CAN0OUT pin to the internal CAN0IN pin. The input value of the external CAN0IN pin is ignored. The external CAN0OUT pin outputs only recessive bits. The CAN0OUT/CAN0IN pins do not need to be connected to the CAN bus or any external device.

Figure 24.33 shows the connection when self test mode 1 is selected.

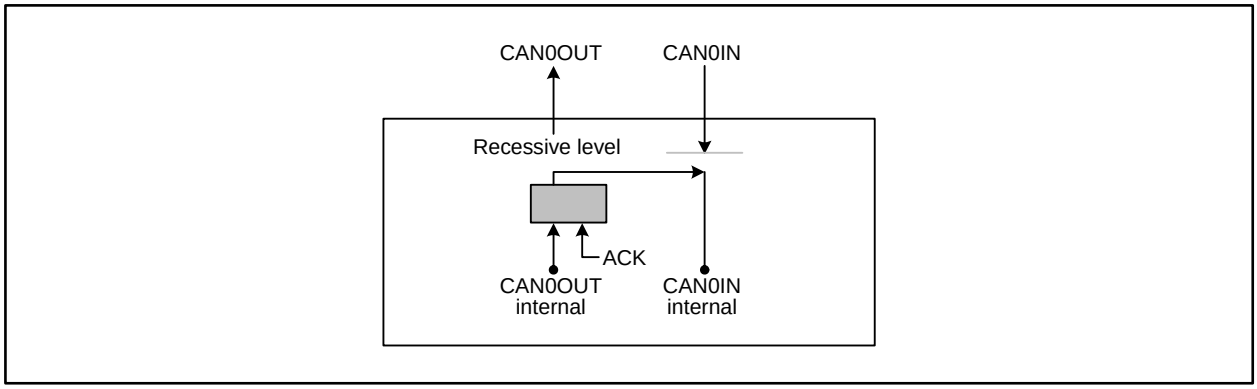


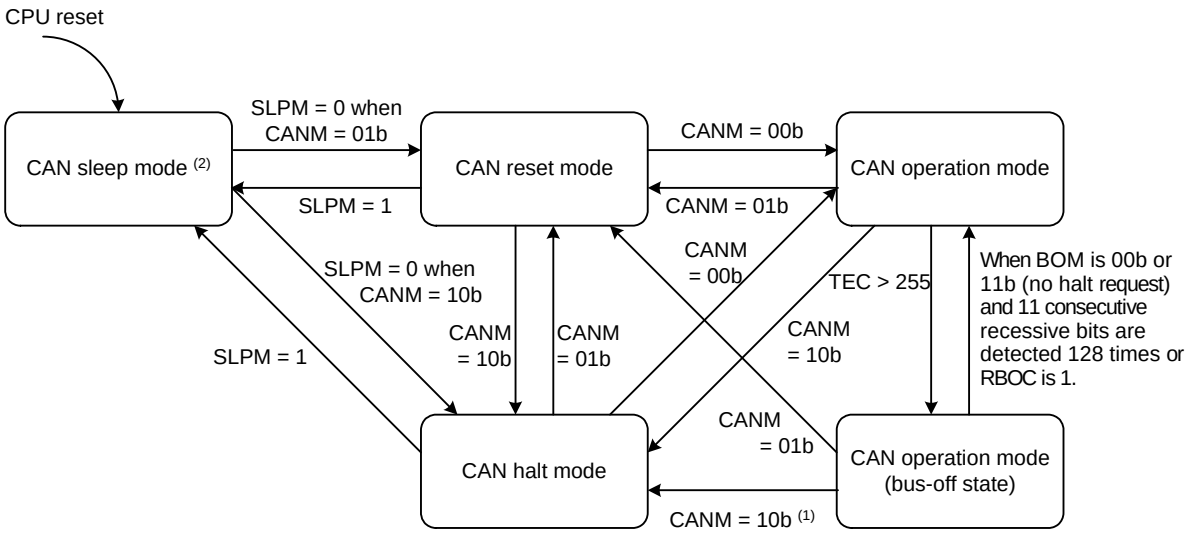
Figure 24.33 Connection when Self Test Mode 1 is Selected

24.2 Operating Mode

The CAN module has the following four operating modes:

- CAN reset mode
- CAN halt mode
- CAN operation mode
- CAN sleep mode

Figure 24.34 shows the transition between CAN operating modes.


CPU reset
CAN sleep mode (2)
SLPM = 0 when CANM = 01b
SLPM = 1
CAN reset mode
CANM = 00b
CANM = 01b
CAN operation mode
SLPM = 0 when CANM = 10b
SLPM = 1
CANM = 10b
CANM = 01b
CANM = 00b
TEC > 255
CANM = 10b
When BOM is 00b or 11b (no halt request) and 11 consecutive recessive bits are detected 128 times or RBOC is 1.
CANM = 01b
CAN halt mode
CAN operation mode (bus-off state)
CANM = 10b (1)

CANM, SLPM, BOM, and RBOC: Bits in the C0CTLR register

Notes:

1. The transition timing from the bus-off state to CAN halt mode depends on the setting of the BOM bit.
   - When the BOM bit is 01b, the state transition timing is immediately after entering the bus-off state.
   - When the BOM bit is 10b, the state transition timing is at the end of the bus-off state.
   - When the BOM bit is 11b, the state transition timing is at the setting of the CANM bit to 10b (CAN halt mode).
2. Write only to the SLPM bit to exit/set CAN sleep mode.

Figure 24.34 Transition between CAN Operating Modes

24.2.1 CAN Reset Mode

CAN reset mode is provided for CAN communication configuration.
When the CANM bit in the C0CTLR register is set to 01b, the CAN module enters CAN reset. Then the RSTST bit in the C0STR register is set to 1. Do not change the CANM bit until the RSTST bit is set to 1. Configure the C0BCR register before exiting CAN reset mode to any other modes.

The following registers are initialized to their reset values after entering CAN reset mode and their initialized values are retained during CAN reset mode:

- C0MCTLj register (j = 0 to 31)
- C0STR register (except bits SLPST and TFST)
- C0EIFR register
- C0RECR register
- C0TECR register
- C0TSR register
- C0MSSR register
- C0MSMR register
- C0RFCR register
- C0TFCR register
- C0TCR register
- C0ECSR register (except EDPM bit)

The previous values of the following registers are retained after entering CAN reset mode.

- C0CLKR register
- C0CTLR register
- C0STR register (bits SLPST and TFST)
- C0MIER register
- C0EIER register
- C0BCR register
- C0CSSR register
- C0ECSR register (EDPM bit only)
- C0MBj register
- Registers C0MKR0 to C0MKR7
- Registers C0FIDCR0 and C0FIDCR1
- C0MKIVLR register
- C0AFSR register
- C0RFPCR register
- C0TFPCR register

24.2.2 CAN Halt Mode

CAN halt mode is used for mailbox configuration and test mode setting.
When the CANM bit in the C0CTLR register is set to 10b, CAN halt mode is selected. Then the HLTST bit in the C0STR register is set to 1. Do not change the CANM bit until the HLTST bit is set to 1.
Refer to Table 24.9 “Operation in CAN Reset Mode and CAN Halt Mode” regarding the state transition conditions when transmitting or receiving.

All registers except bits RSTST, HLTST, and SLPST in the C0STR register remain unchanged when the CAN module enters CAN halt mode.
Do not change registers C0CLKR, C0CTLR (except bits CANM and SLPM), and C0EIER in CAN halt mode. The C0BCR register can be changed in CAN halt mode only when listen only mode is selected to use for automatic bit rate detection.

Table 24.9 Operation in CAN Reset Mode and CAN Halt Mode

Mode	Receiver	Transmitter	Bus-off
CAN reset mode	CAN module enters CAN reset mode without waiting for the end of message reception.	CAN module enters CAN reset mode after waiting for the end of message transmission. (1, 4)	CAN module enters CAN reset mode without waiting for the end of bus-off recovery.
CAN halt mode	CAN module enters CAN halt mode after waiting for the end of message reception. (2, 3)	CAN module enters CAN halt mode after waiting for the end of message transmission. (1, 4)	[When the BOM bit is 00b] A halt request from a program will be acknowledged only after bus-off recovery. [When the BOM bit is 01b] CAN module enters automatically to CAN halt mode without waiting for the end of bus-off recovery (regardless of a halt request from a program). [When the BOM bit is 10b] CAN module enters automatically to CAN halt mode after waiting for the end of bus-off recovery (regardless of a halt request from a program). [When the BOM bit is 11b] CAN module enters CAN halt mode (without waiting for the end of bus-off recovery) if a halt is requested by a program during bus-off.

BOM bit: Bit in the C0CTLR register
Notes:
1. If several messages are requested to be transmitted, mode transition occurs after the completion of the first transmission. In a case that the CAN reset mode is being requested during suspend transmission, mode transition occurs when the bus is idle, the next transmission ends, or the CAN module becomes a receiver.
2. If the CAN bus is locked at the dominant level, the program can detect this state by monitoring the BLIF bit in the C0EIFR register.
3. If a CAN bus error occurs during reception after CAN halt mode is requested, the CAN mode transits to CAN halt mode.
4. If a CAN bus error or arbitration lost occurs during transmission after CAN reset mode or CAN halt mode is requested, the CAN mode transits to the requested CAN mode.

24.2.3 CAN Sleep Mode

CAN sleep mode is used for reducing current consumption by stopping the clock supply to the CAN module. After MCU hardware reset or software reset, the CAN module starts from CAN sleep mode.

When the SLPM bit in the C0CTLR register is set to 1, the CAN module enters CAN sleep mode. Then the SLPST bit in the C0STR register is set to 1. Do not change the value of the SLPM bit until the bit is set to 1. The other registers remain unchanged when the MCU enters CAN sleep mode.

Write to the SLPM bit in CAN reset mode and CAN halt mode. Do not change any other registers (except the SLPM bit) during CAN sleep mode. Read operation is still allowed.
When the SLPM bit is set to 0, the CAN module is released from CAN sleep mode. When the CAN module exits CAN sleep mode, the other registers remain unchanged.

24.2.4 CAN Operation Mode (Excluding Bus-off State)

CAN operation mode is used for CAN communication.
When the CANM bit in the C0CTLR register is set to 00b, the CAN module enters CAN operation mode. Then bits RSTST and HLTST in the C0STR register are set to 0. Do not change the value of the CANM bit until these bits are set to 0.

If 11 consecutive recessive bits are detected after entering CAN operation mode, the CAN module is in the following states:

- The CAN module becomes an active node on the network that enables transmission and reception of CAN messages.
- Error monitoring of the CAN bus, such as receive and transmit error counters, is performed.

During CAN operation mode, the CAN module may be in one of the following three sub-modes, depending on the status of the CAN bus:

- Idle mode: Transmission or reception is not being performed.
- Receive mode: A CAN message sent by another node is being received.
- Transmit mode: A CAN message is being transmitted. The CAN module may receive its own message simultaneously when self test mode 0 (TSTM bit in the C0TCR register = 10b) or self test mode 1 (TSTM bit = 11b) is selected.

Figure 24.35 shows the sub mode in CAN operation mode.

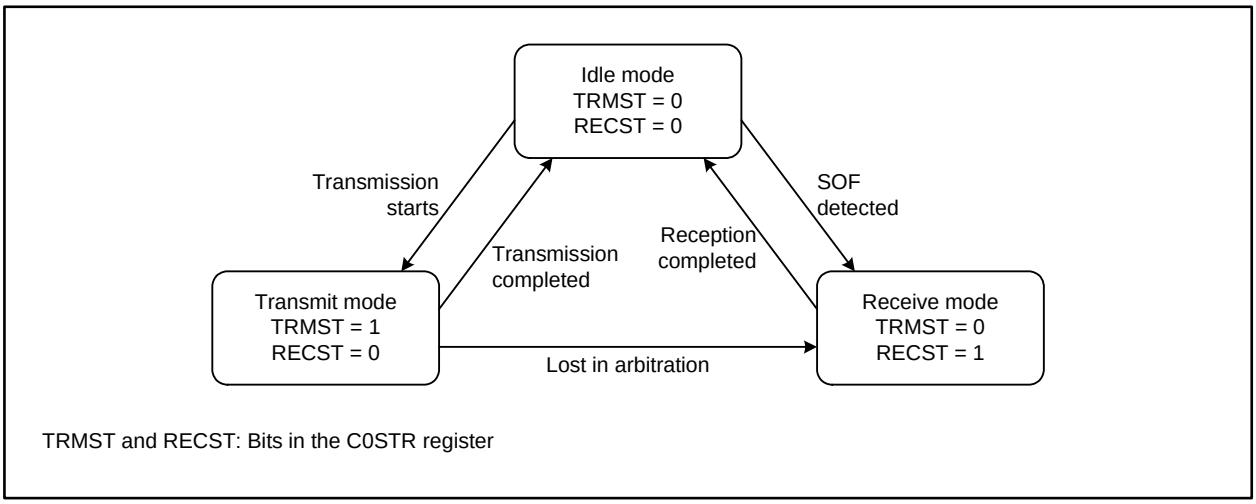


Figure 24.35 Sub Mode in CAN Operation Mode

24.2.5 CAN Operation Mode (Bus-off State)

The CAN module enters the bus-off state according to the increment/decrement rules for the transmit/error counters in the CAN Specifications.
The following cases apply when recovering from the bus-off state. When the CAN module is in bus-off state, the values of the associated registers, except registers C0STR, C0EIFR, C0RECR, C0TECR, and C0TSR, remain unchanged.

(1) When the BOM bit in the C0CTLR register is 00b (normal mode)

The CAN module enters the error-active state after it has completed the recovery from the bus-off state and CAN communication is enabled. The BORIF bit in the C0EIFR register is set to 1 (bus-off recovery detected) at this time.

(2) When the RBOC bit in the C0CTLR register is set to 1 (forcible return from bus-off)

The CAN module enters the error-active state when it is in bus-off state and the RBOC bit is set to 1. CAN communication is enabled again after 11 consecutive recessive bits are detected. The BORIF bit is not set to 1 at this time.

(3) When the BOM bit is 01b (entry to CAN halt mode automatically at bus-off entry)

The CAN module enters CAN halt mode when it reaches the bus-off state. The BORIF bit is not set to 1 at this time.

(4) When the BOM bit is 10b (entry to CAN halt mode automatically at bus-off end)

The CAN module enters CAN halt mode when it has completed the recovery from bus-off. The BORIF bit is set to 1 at this time.

(5) When the BOM bit is 11b (entry to CAN halt mode by a program) and the CANM bit in the C0CTLR register is set to 10b (CAN halt mode) during the bus-off state

The CAN module enters CAN halt mode when it is in bus-off state and the CANM bit is set to 10b (CAN halt mode). The BORIF bit is not set to 1 at this time.
If the CANM bit is not set to 10b during bus-off, the same behavior as (1) applies.

24.3 CAN Communication Speed Configuration

The following description explains about the CAN communication speed configuration.

24.3.1 CAN Clock Configuration

This group has a CAN clock selector.
The CAN clock can be configured by setting the CCLKS bit in the C0CLKR register and the BRP bit in the C0BCR register.
Figure 24.36 shows the block diagram of CAN clock generator.

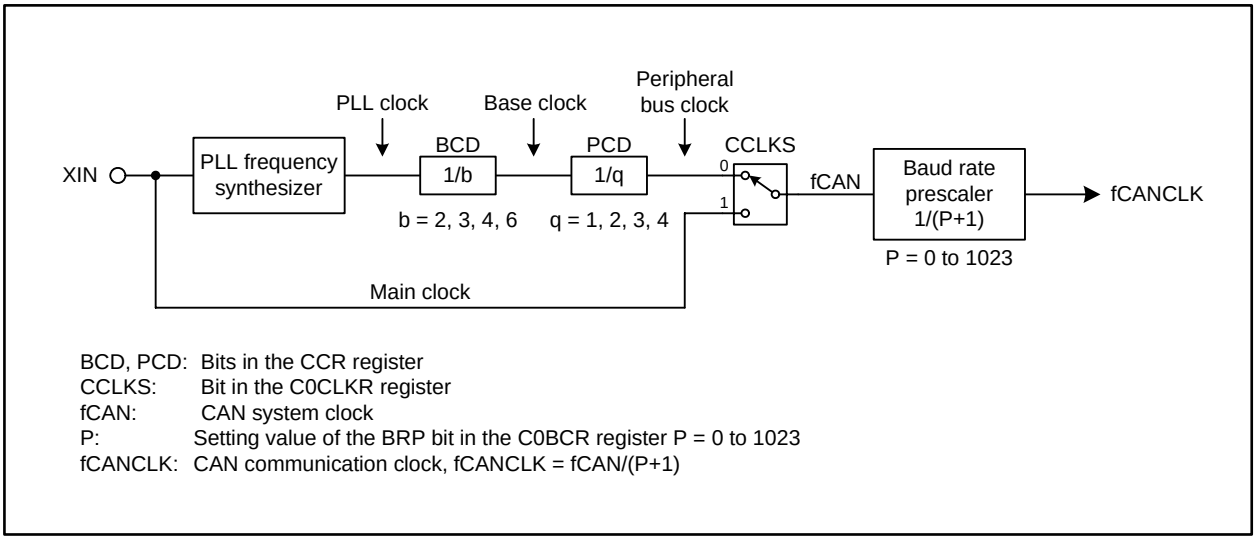


Figure 24.36 Block Diagram of CAN Clock Generator

24.3.2 Bit Timing Configuration

The bit time is a single bit time for transmitting/receiving a message and consists of the following three segments.
Figure 24.37 shows the bit timing.

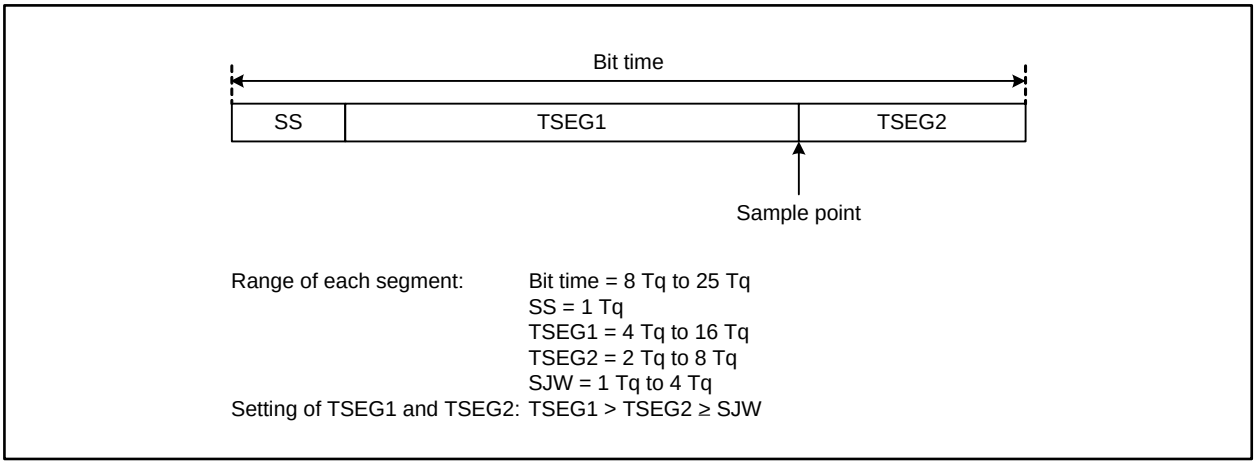

Bit time
SS
TSEG1
TSEG2
Sample point
Range of each segment: Bit time = 8 Tq to 25 Tq
SS = 1 Tq
TSEG1 = 4 Tq to 16 Tq
TSEG2 = 2 Tq to 8 Tq
SJW = 1 Tq to 4 Tq
Setting of TSEG1 and TSEG2: TSEG1 > TSEG2 ≥ SJW

Figure 24.37 Bit Timing

24.3.3 Bit rate

The bit rate depends on the CAN clock (fCAN), the division value of the baud rate prescaler, and the number of Tq of one bit time.

$$\text{Bit rate[bps]} = \frac{fCAN}{\text{Baud rate prescaler division value}\ (1) \times \text{number of Tq of one bit time}} = \frac{fCANCLK}{\text{Number of Tq of one bit time}}$$

Note:
1. Division value of the baud rate prescaler = P + 1 (P = 0 to 1023)
   P: Setting value of the BRP bit in the C0BCR register

Table 24.10 lists bit rate examples.

Table 24.10 Bit Rate Examples

fCAN	32 MHz		24 MHz		20 MHz		16 MHz		8 MHz	
Bit Rate	No. of Tq	P+1	No. of Tq	P+1	No. of Tq	P+1	No. of Tq	P+1	No. of Tq	P+1
1 Mbps	8 Tq	4	8 Tq	3	10 Tq	2	8 Tq	2	8 Tq	1
	16 Tq	2			20 Tq	1	16 Tq	1		
500 kbps	8 Tq	8	8 Tq	6	10 Tq	4	8 Tq	4	8 Tq	2
	16 Tq	4	16 Tq	3	20 Tq	2	16 Tq	2	16 Tq	1
250 kbps	8 Tq	16	8 Tq	12	10 Tq	8	8 Tq	8	8 Tq	4
	16 Tq	8	16 Tq	6	20 Tq	4	16 Tq	4	16 Tq	2
83.3 kbps	8 Tq	48	8 Tq	36	8 Tq	30	8 Tq	24	8 Tq	12
	16 Tq	24	16 Tq	18	10 Tq	24	16 Tq	12	16 Tq	6
					16 Tq	15				
					20 Tq	12				
33.3 kbps	8 Tq	120	8 Tq	90	8 Tq	75	8 Tq	60	8 Tq	30
	10 Tq	96	10 Tq	72	10 Tq	60	10 Tq	48	10 Tq	24
	16 Tq	60	16 Tq	45	20 Tq	30	16 Tq	30	16 Tq	15
	20 Tq	48	20 Tq	36			20 Tq	24	20 Tq	12

24.4 Mailbox and Mask Register Structure

There are 32 mailboxes with the same structure.
Figure 24.38 shows the structure of C0MBj register (j = 0 to 31).

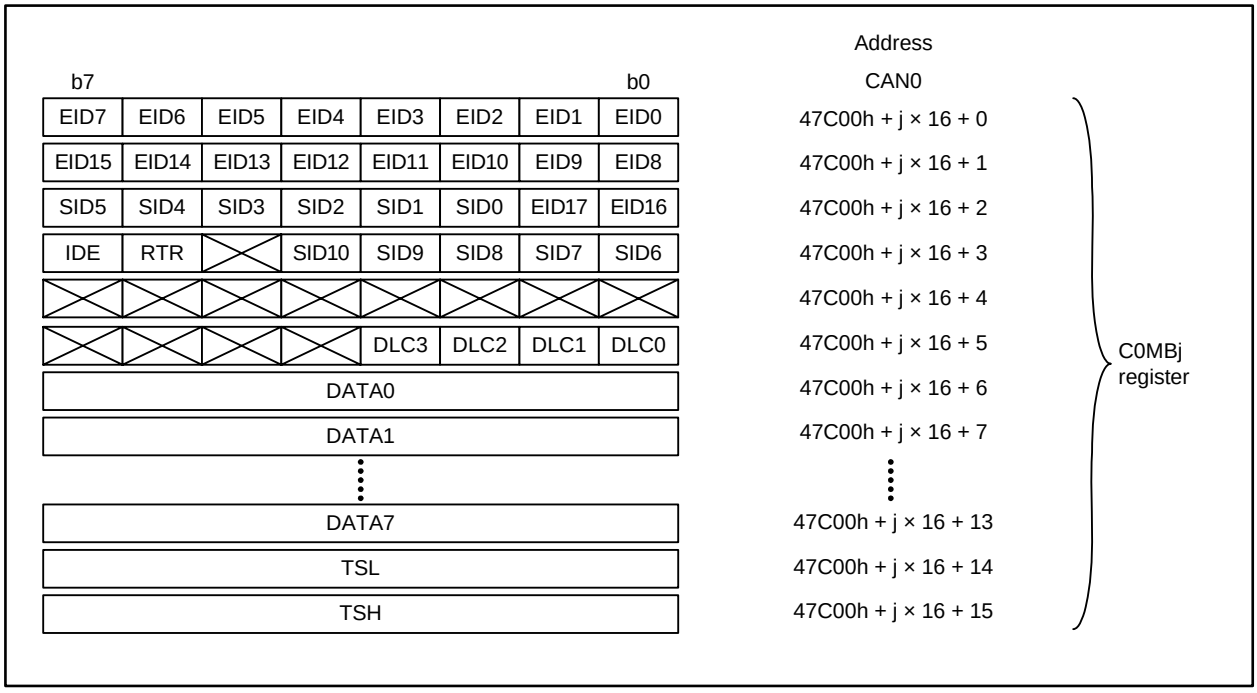

Address
b7
b0
CAN0
EID7 EID6 EID5 EID4 EID3 EID2 EID1 EID0
47C00h + j × 16 + 0
EID15 EID14 EID13 EID12 EID11 EID10 EID9 EID8
47C00h + j × 16 + 1
SID5 SID4 SID3 SID2 SID1 SID0 EID17 EID16
47C00h + j × 16 + 2
IDE RTR SID10 SID9 SID8 SID7 SID6
47C00h + j × 16 + 3
47C00h + j × 16 + 4
DLC3 DLC2 DLC1 DLC0
47C00h + j × 16 + 5
DATA0
47C00h + j × 16 + 6
DATA1
47C00h + j × 16 + 7
DATA7
47C00h + j × 16 + 13
TSL
47C00h + j × 16 + 14
TSH
47C00h + j × 16 + 15
C0MBj register

Figure 24.38 Structure of C0MBj Register (j = 0 to 31)

There are 8 mask registers with the same structure.
Figure 24.39 shows the structure of C0MKRk Register (k = 0 to 7).

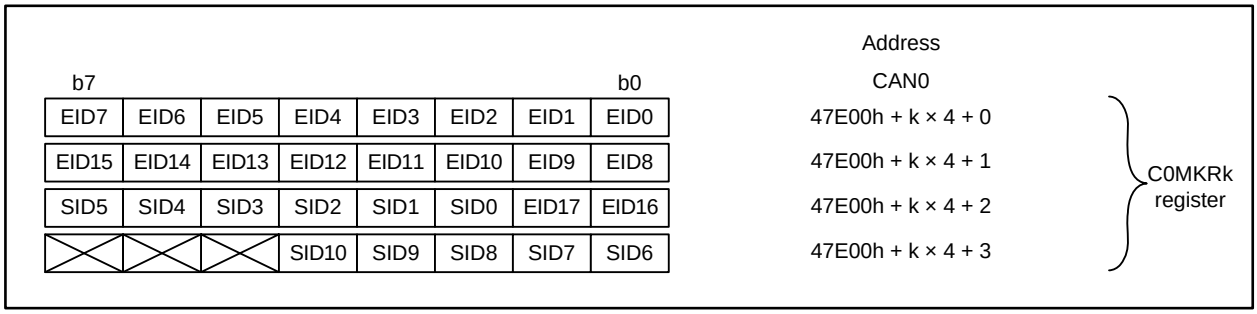

Address
b7
b0
CAN0
EID7 EID6 EID5 EID4 EID3 EID2 EID1 EID0
47E00h + k × 4 + 0
EID15 EID14 EID13 EID12 EID11 EID10 EID9 EID8
47E00h + k × 4 + 1
SID5 SID4 SID3 SID2 SID1 SID0 EID17 EID16
47E00h + k × 4 + 2
SID10 SID9 SID8 SID7 SID6
47E00h + k × 4 + 3
C0MKRk register

Figure 24.39 Structure of C0MKRk Register (k = 0 to 7)

There are 2 FIFO received ID compare registers with the same structure.
Figure 24.40 shows the structure of C0FIDCRn Register (n = 0, 1).

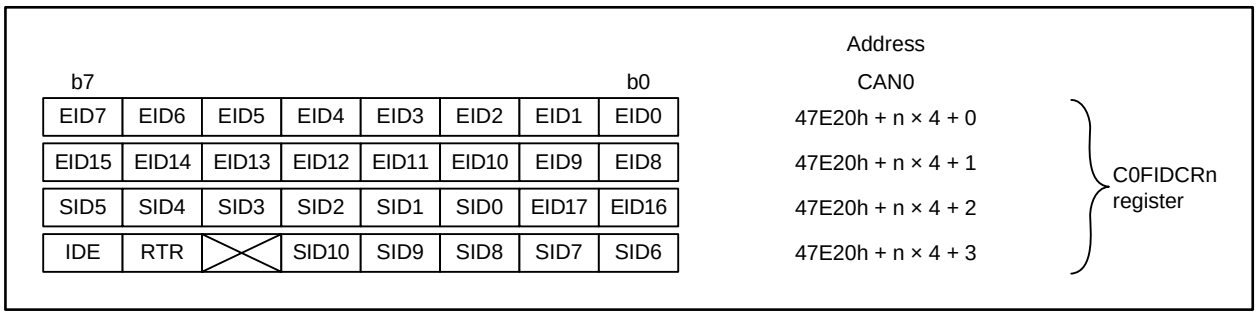

b7
b0
Address
CAN0
EID7 EID6 EID5 EID4 EID3 EID2 EID1 EID0
47E20h + n × 4 + 0
EID15 EID14 EID13 EID12 EID11 EID10 EID9 EID8
47E20h + n × 4 + 1
SID5 SID4 SID3 SID2 SID1 SID0 EID17 EID16
47E20h + n × 4 + 2
IDE RTR SID10 SID9 SID8 SID7 SID6
47E20h + n × 4 + 3
C0FIDCRn register

Figure 24.40 Structure of C0FIDCRn Register (n = 0, 1)

24.5 Acceptance Filtering and Masking Function

Acceptance filtering allows the user to receive messages with a specified range of multiple IDs for mailboxes.
Registers C0MKR0 to C0MKR7 can perform masking of the standard ID and the extended ID of 29 bits.

- The C0MKR0 register corresponds to mailboxes [0] to [3].
- The C0MKR1 register corresponds to mailboxes [4] to [7].
- The C0MKR2 register corresponds to mailboxes [8] to [11].
- The C0MKR3 register corresponds to mailboxes [12] to [15].
- The C0MKR4 register corresponds to mailboxes [16] to [19].
- The C0MKR5 register corresponds to mailboxes [20] to [23].
- The C0MKR6 register corresponds to mailboxes [24] to [27] in normal mailbox mode, and receive FIFO mailboxes [28] to [31] in FIFO mailbox mode.
- The C0MKR7 register corresponds to mailboxes [28] to [31] in normal mailbox mode, and receive FIFO mailboxes [28] to [31] in FIFO mailbox mode.

The C0MKIVLR register disables acceptance filtering individually for each mailbox.

The IDE bit in the C0MBj register (j = 0 to 31) is enabled when the IDFM bit in the C0CTLR register is 10b (mixed ID mode).
The RTR bit in the C0MBj register selects a data frame or a remote frame.

In FIFO mailbox mode, normal mailboxes (mailboxes [0] to [23]) use the single corresponding register among registers C0MKR0 to C0MKR5 for acceptance filtering. Receive FIFO mailboxes (mailboxes [28] to [31]) use two registers C0MKR6 and C0MKR7 for the acceptance filtering.
Also, the receive FIFO uses two registers C0FIDCR0 and C0FIDCR1 for ID comparison. Bits EID, SID, RTR, and IDE in registers C0MB28 to C0MB31 for the receive FIFO are disabled. As acceptance filtering depends on the result of two ID-mask sets, two ranges of IDs can be received into the receive FIFO.
The C0MKIVLR register is disabled for the receive FIFO.

If both setting of standard ID and extended ID are set in the IDE bits in registers C0FIDCR0 and C0FIDCR1 individually, both ID formats are received.
If both setting of data frame and remote frame are set in the RTR bits in registers C0FIDCR0 and C0FIDCR1 individually, both data and remote frames are received.
When combination with two ranges of IDs is not necessary, set the same mask value and the same ID into both of the FIFO ID/mask register sets.

Figure 24.41 shows the correspondence of mask registers to mailboxes, and Figure 24.42 shows acceptance filtering.

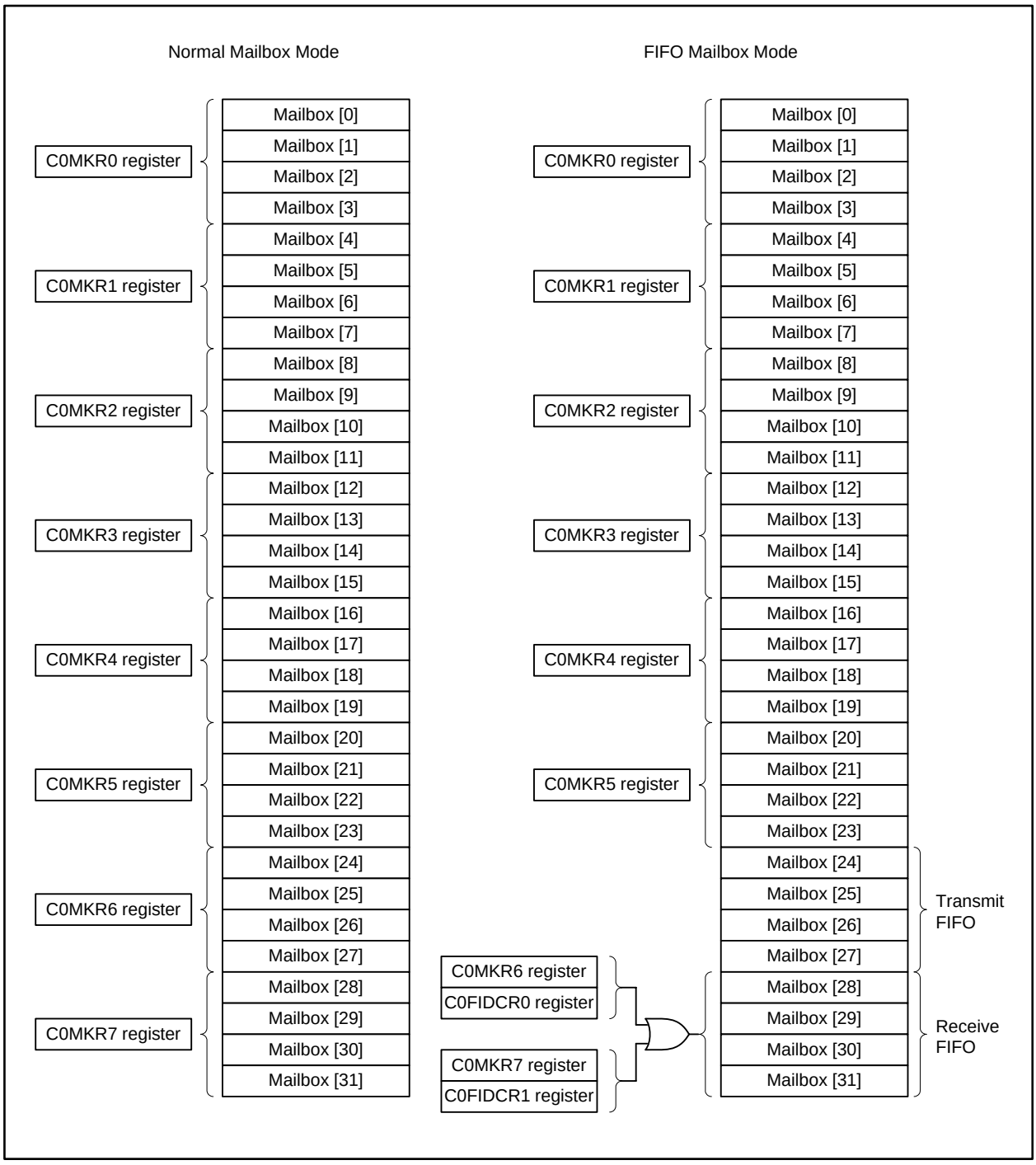


Figure 24.41 Correspondence of Mask Registers to Mailboxes

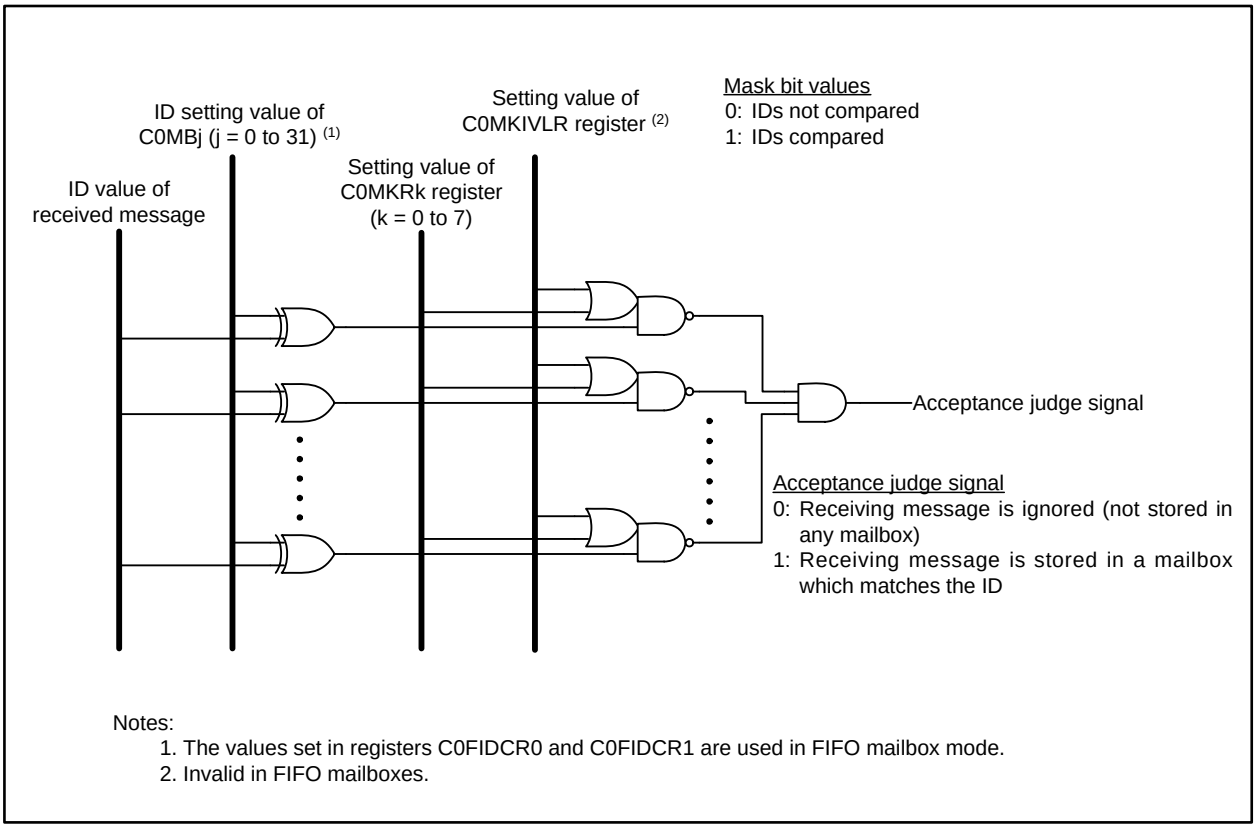

ID setting value of
C0MBj (j = 0 to 31) (1)
Setting value of
C0MKIVLR register (2)
Mask bit values
0: IDs not compared
1: IDs compared
ID value of
received message
Setting value of
C0MKRk register
(k = 0 to 7)
Acceptance judge signal
Acceptance judge signal
0: Receiving message is ignored (not stored in any mailbox)
1: Receiving message is stored in a mailbox which matches the ID
Notes:
1. The values set in registers C0FIDCR0 and C0FIDCR1 are used in FIFO mailbox mode.
2. Invalid in FIFO mailboxes.

Figure 24.42 Acceptance Filtering

24.6 Reception and Transmission

Table 24.11 lists the CAN communication mode configuration.

Table 24.11 Configuration for CAN Reception Mode and Transmission Mode

TRMREQ	RECREQ	ONESHOT	Communication Mode of Mailbox
0	0	0	Mailbox disabled or transmission being aborted.
0	0	1	Configurable only when transmission or reception from a mailbox (programmed in one-shot mode) is aborted.
0	1	0	Configured as a receive mailbox for a data frame or a remote frame.
0	1	1	Configured as a one-shot receive mailbox for a data frame or a remote frame.
1	0	0	Configured as a transmit mailbox for a data frame or a remote frame.
1	0	1	Configured as a one-shot transmit mailbox for a data frame or a remote frame.
1	1	0	Do not set.
1	1	1	Do not set.

TRMREQ, RECREQ, ONESHOT: Bits in the C0MCTLj register (j = 0 to 31)

When a mailbox is configured as a receive mailbox or a one-shot receive mailbox, note the following:

(1) Before a mailbox is configured as a receive mailbox or a one-shot receive mailbox, set the C0MCTLj register (j = 0 to 31) to 00h.

(2) A received message is stored into the first mailbox that matches the condition according to the result of receive mode configuration and acceptance filtering. Upon deciding a mailbox which stores the received message, the mailbox with the smaller number has higher priority.

(3) In CAN operation mode, when a CAN module transmits a message whose ID matches with the ID/mask set of a mailbox configured to receive messages, the CAN module never receives the transmitted data. In self test mode, however, the CAN module may receive its transmitted data. In this case, the CAN module sends an ACK.

When a mailbox is configured as a transmit mailbox or a one-shot transmit mailbox, note the following:

(1) Before a mailbox is configured as a transmit mailbox or one-shot transmit mailbox, ensure that the C0MCTLj register is 00h and that there is no pending abort process.

24.6.1 Reception

Figure 24.43 shows an operation example of data frame reception in overwrite mode.
This example shows the operation of overwriting the first message when the CAN module receives two consecutive CAN messages that matches the receiving conditions of the C0MCTL0 register.

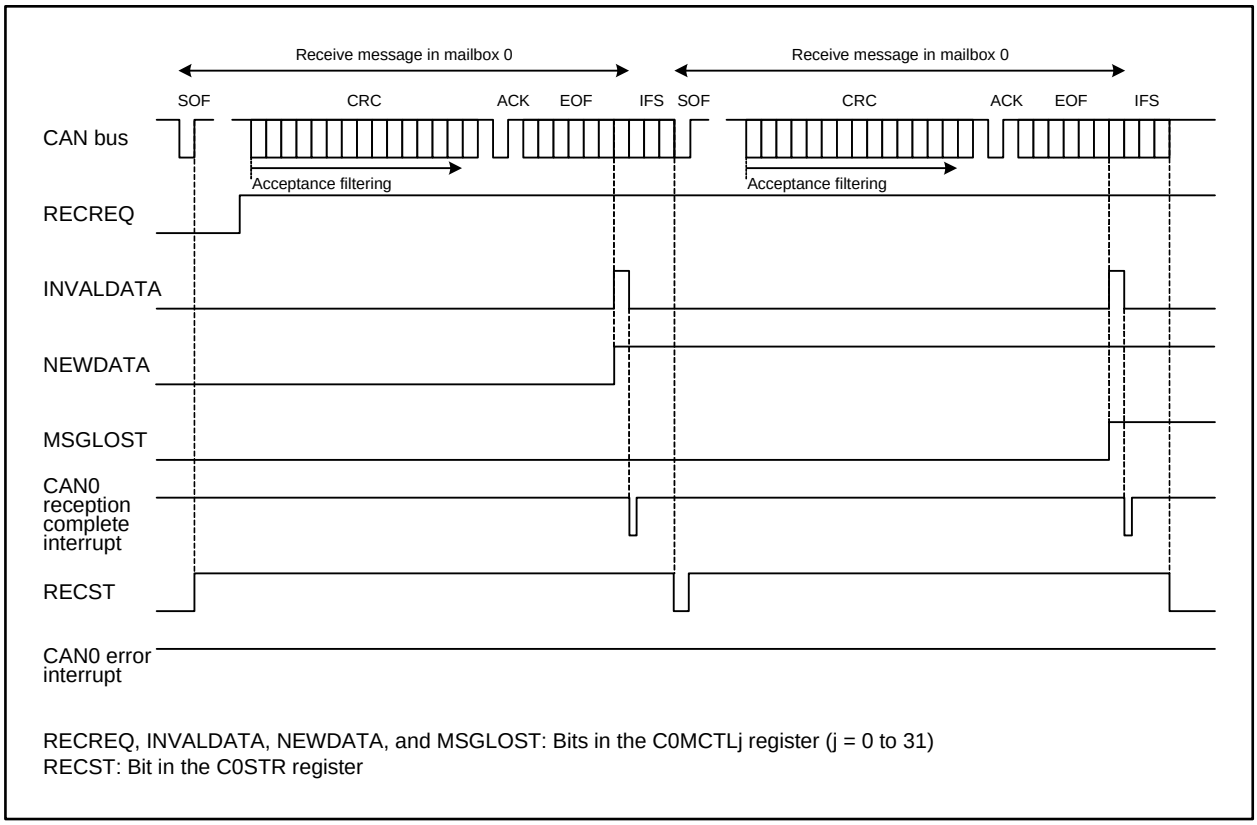


Figure 24.43 Operation Example of Data Frame Reception in Overwrite Mode

(1) When a SOF is detected on the CAN bus, the RECST bit in the C0STR register is set to 1 (reception in progress) if the CAN module has no message ready to start transmission.
(2) The acceptance filter procedure starts at the beginning of the CRC field to select the receive mailbox.
(3) After a message has been received, the NEWDATA bit in the C0MCTLj register (j = 0 to 31) for the receive mailbox is set to 1 (new data being updated/stored in the mailbox). The INVALDATA bit in the C0MCTLj register is set to 1 (message is being updated) at the same time, and then the INVALDATA bit is set to 0 (message valid) again after the complete message is transferred to the mailbox.
(4) When the interrupt enable bit in the C0MIER register for the receive mailbox is 1 (interrupt enabled), the CAN0 reception complete interrupt request is generated. This interrupt is generated when the INVALDATA bit is set to 0.
(5) After reading the message from the mailbox, the NEWDATA bit needs to be set to 0 by a program.
(6) In overwrite mode, if the next CAN message has been received into a mailbox whose NEWDATA bit is still set to 1, the MSGLOST bit in the C0MCTLj register is set to 1 (message has been overwritten). The new received message is transferred to the mailbox. The CAN0 reception complete interrupt request is generated the same as in (4).

Figure 24.44 shows the operational example of data frame reception in overrun mode.
This example shows the operation of overrunning the second message when the CAN module receives two consecutive CAN messages that matches the receiving conditions of the C0MCTL0 register.

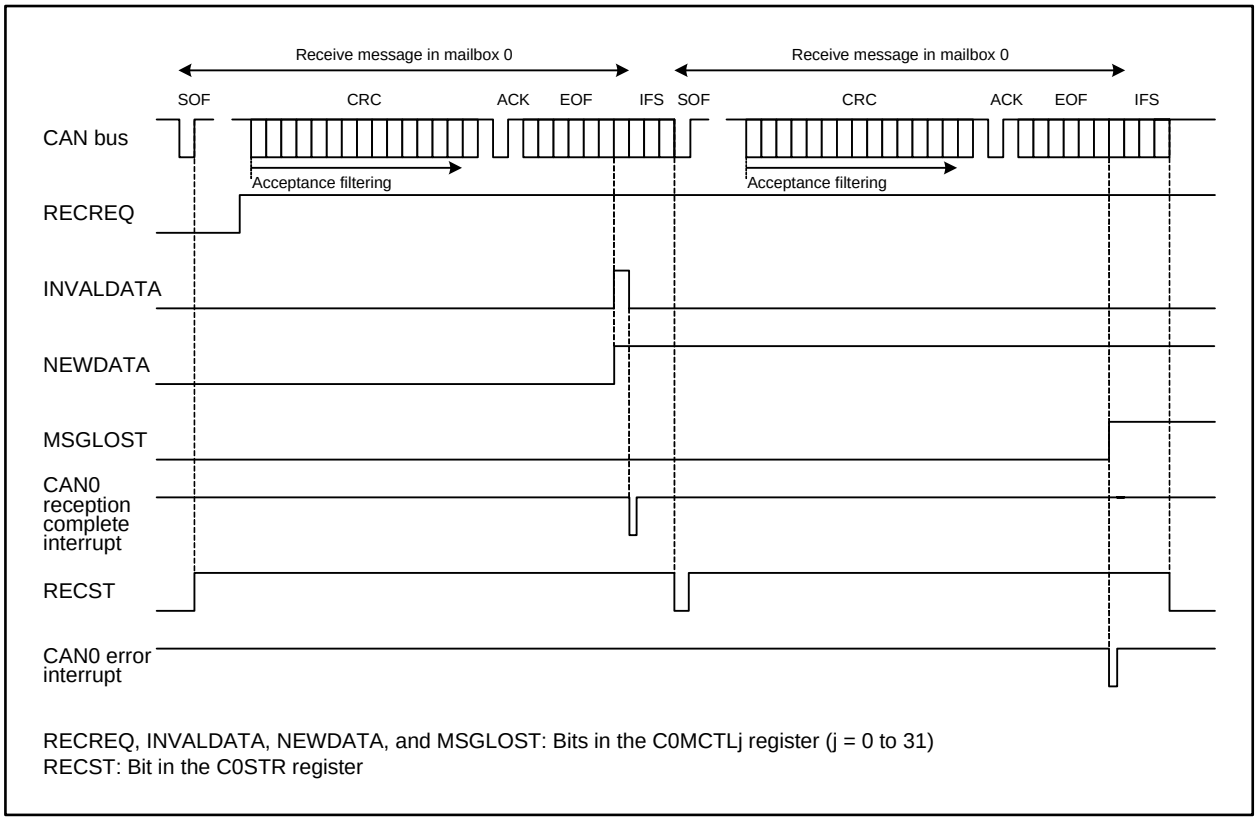


Figure 24.44 Operation Example of Data Frame Reception in Overrun Mode

(1) to (5) are the same as overwrite mode.

(6) In overrun mode, if the next message has been received before the NEWDATA bit is set to 0, the MSGLOST bit in the C0MCTLj register (j = 0 to 31) is set to 1 (message has been overrun). The new received message is discarded and a CAN0 error interrupt request is generated if the corresponding interrupt enable bit in the C0EIER register is set to 1 (interrupt enabled).

24.6.2 Transmission

Figure 24.45 shows an operation example of data frame transmission. This example shows the operation of transmitting messages that has been set in registers C0MCTL0 and C0MCTL1.

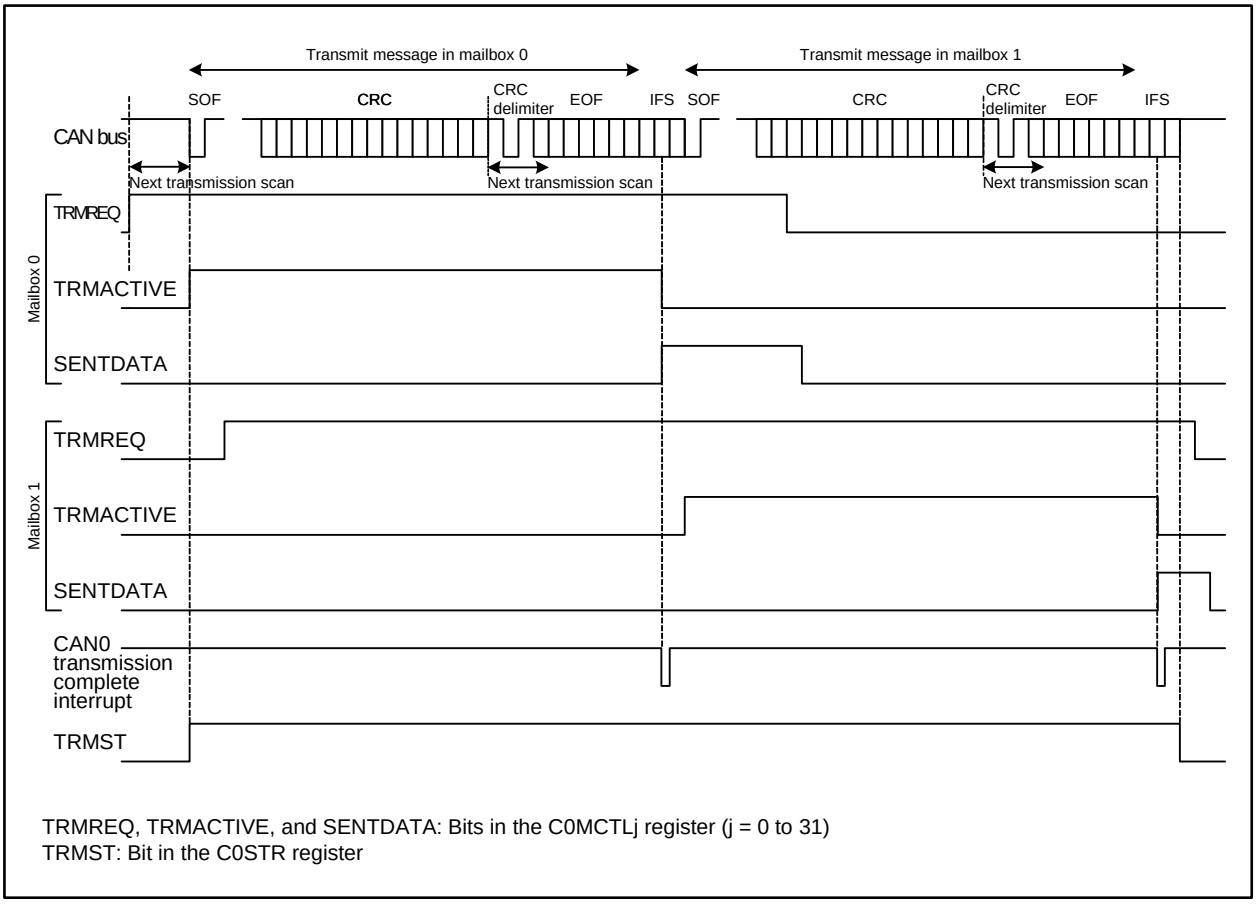


Figure 24.45 Operation Example of Data Frame Transmission

(1) When a TRMREQ bit in the C0MCTLj register (j = 0 to 31) is set to 1 (transmit mailbox) in bus-idle state, the mailbox scan procedure starts to decide the highest-priority mailbox for transmission. Once the transmit mailbox is decided, the TRMACTIVE bit in the C0MCTLj register is set to 1 (from when a transmission request is received until transmission is completed, or an error/arbitration lost has occurred), the TRMST bit in the C0STR register is set to 1 (transmission in progress), and the CAN module starts transmission. (1)

(2) If other TRMREQ bits are set, the transmission scan procedure starts with the CRC delimiter for the next transmission.

(3) If transmission is completed without losing arbitration, the SENTDATA bit in the C0MCTLj register is set to 1 (transmission completed) and the TRMACTIVE bit is set to 0 (transmission is pending, or no transmission request). If the interrupt enable bit in the C0MIER register is 1 (interrupt enabled), the CAN0 transmission complete interrupt request is generated.

(4) When requesting the next transmission from the same mailbox, set bits SENDTDATA and TRMREQ to 0, then set the TRMREQ bit to 1 after checking that bits SENDTDATA and TRMREQ have been set to 0.

Note:

1. If arbitration is lost after the CAN module starts transmission, the TRMACTIVE bit is set to 0. The transmission scan procedure is performed again to search for the highest-priority transmit mailbox from the beginning of the CRC delimiter. If an error occurs either during transmission or following the loss of arbitration, the transmission scan procedure is performed again from the start of the error delimiter to search for the highest-priority transmit mailbox.

24.7 CAN Interrupt

The CAN module provides the following CAN interrupts:

- CAN0 wakeup interrupt
- CAN0 reception complete interrupt
- CAN0 transmission complete interrupt
- CAN0 receive FIFO interrupt
- CAN0 transmit FIFO interrupt
- CAN0 error interrupt

There are eight types of interrupt sources for the CAN0 error interrupts. These sources can be determined by checking the C0EIFR register.

- Bus error
- Error-warning
- Error-passive
- Bus-off entry
- Bus-off recovery
- Receive overrun
- Overload frame transmission
- Bus lock

25. I/O Pins

Each pin of the MCU functions as a programmable I/O port or an I/O pin for internal peripheral functions. These functions can be switched by the function select registers. The pull-up resistors are enabled for every group of four pins. However, a pull-up resistor is separated from other peripheral functions even if it is enabled, when a pin functions as an output pin.
Figure 25.1 shows a block diagram of typical I/O pin.

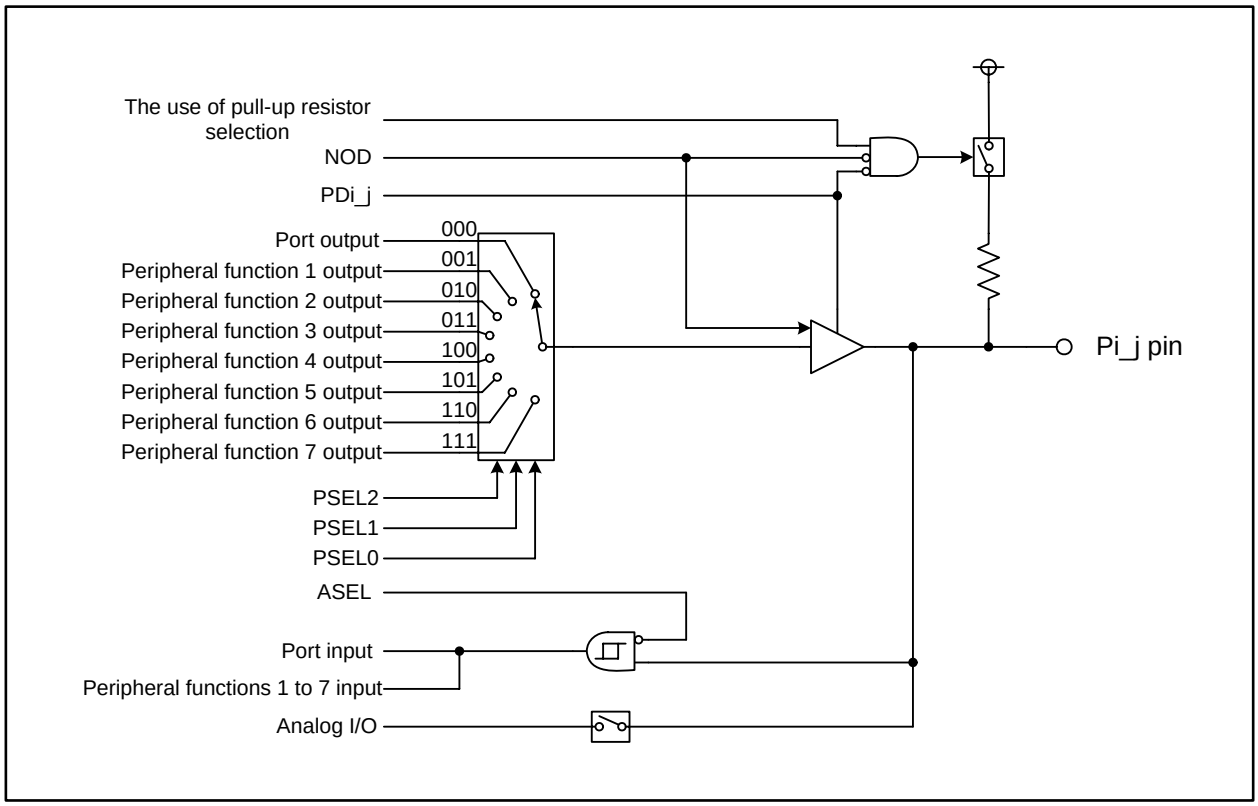


Figure 25.1 Typical I/O Pin Block Diagram (i = 0 to 15; j = 0 to 7)

The registers to control I/O pins are as follows: port Pi direction register (PDi register), output function select register, and pull-up control register. The PDi register selects input or output state of pins. The output function select register which selects an output function consists of bits PSEL2 to PSEL0, NOD, and ASEL. Bits PSEL2 to PSEL0 are to select a function as programmable I/O or peripheral function output (except analog output). The NOD bit is to select the N-channel open drain output for a pin. The ASEL bit enables to prevent the increase in power consumption of input buffer due to an intermediate potential when a pin functions as an analog I/O pin. The pull-up control register enables/disables the pull-up resistors.
To use a pin as analog I/O pin, the PDi_j bit should be set to 0 (input) and bits PSEL2 to PSEL0 should be set to 000b and the ASEL bit should be set to 1.
The input-only port P8_5, which shares a pin with the $\overline{\text{NMI}}$ has neither bit 5 of the function select register nor the PDi register. The port P14_1 also functions as input-only port. Bit 1 of the function select register and the PDi register is assigned for reserved bit. The port P9 is protected from unexpected write accesses by the PRC2 bit in the PRCR register. (Refer to 9. “Protection”)

25.1 Port Pi Direction Register (PDi Register, i = 0 to 15)

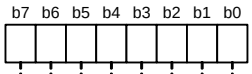
The PDi register selects input or output state of pins. Each bit in this register corresponds to a respective pin.

Figure 25.2 shows the PDi register.

No register bit is provided for the P8_5. For the P14_1, a reserved bit is provided.

The PD9 register is protected from unexpected write accesses by setting the PRC2 bit in the PRCR register. (Refer to 9. “Protection”)

Port Pi Direction Register (i = 0 to 15)



Symbol	Address	Reset Value
PD0 to PD3	03C2h, 03C3h, 03C6h, 03C7h	0000 0000b
PD4 to PD7	03CAh, 03CBh, 03CEh, 03CFh	0000 0000b
PD8 (1)	03D2h	00X0 0000b
PD9 (2), PD10	03D3h, 03D6h	0000 0000b
PD11 (1)	03D7h	XXX0 0000b
PD12, PD13	03DAh, 03DBh	0000 0000b
PD14 (1, 3)	03DEh	X000 0000b
PD15	03DFh	0000 0000b

Bit Symbol	Bit Name	Function	RW
PDi_0	Port Pi_0 Direction Bit (3)	0: Input port 1: Output port	RW
PDi_1	Port Pi_1 Direction Bit (3)	0: Input port 1: Output port	RW
PDi_2	Port Pi_2 Direction Bit (3)	0: Input port 1: Output port	RW
PDi_3	Port Pi_3 Direction Bit	0: Input port 1: Output port	RW
PDi_4	Port Pi_4 Direction Bit	0: Input port 1: Output port	RW
PDi_5	Port Pi_5 Direction Bit (1)	0: Input port 1: Output port	RW
PDi_6	Port Pi_6 Direction Bit (1)	0: Input port 1: Output port	RW
PDi_7	Port Pi_7 Direction Bit (1)	0: Input port 1: Output port	RW

Notes:
1. The PD8_5 bit in the PD8 register, bits PD11_5 to PD11_7 in the PD11 register and the PD14_7 bit in the PD14 register are unavailable on this MCU. These bits should be written with 0 and read as undefined value.
2. The PD9 register should be written immediately after the PRC2 bit in the PRC2 register is set to 1 (write enabled). Any interrupt or DMA transfer should not be generated between these two instructions.
3. Bits PD14_0 to PD14_2 in the PD14 register are reserved. These bits should be written with 0.

Figure 25.2 Registers PD0 to PD15

Ports P11_0 to P11_2 are used for the communication with audio DSP. Set bits PD11_0 and PD11_1 to 1 (output port) and the PD11_2 bit to 0 (input port). Ports P15_4 to P15_7 are used for the control of audio CODEC. Set bits PD15_4 to PD15_7 to 1 (output port).

Ports P10_0 to P10_7 are connected to general purpose I/O ports of the digital audio processor DP_0 to DP_7 in a wired-OR configuration, respectively. Do not set both ports using the same pin as output ports.

25.2 Output Function Select Register

This register selects an output function of either the programmable I/O port or a peripheral function if these two functions share a pin. Regarding input function, every connected peripheral functions obtain input signals irrespective of this register setting.

The output function select register consists of bits PSEL2 to PSEL0, NOD, and ASEL. Bits PSEL2 to PSEL0 select a function as programmable I/O or peripheral function output (except analog output). The NOD bit is to select the N-channel open drain output. The ASEL bit enables to prevent the increase in power consumption due to an intermediate potential generated when a pin functions as an analog I/O pin.

Table 25.1 shows the peripheral functions assigned for each combination of bits PSEL2 to PSEL0 and Figure 25.3 to Figure 25.18 show the function select registers.

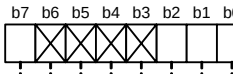
Note that ports P8_5 and P14_1 (input only) have no output function select registers.

The P9_iS register is protected from unexpected write accesses by setting the PRC2 bit in the PRCR register (Refer to 9. “Protection”)

Table 25.1 Peripheral Function Assignment

Bits PSEL2 to PSEL0	Peripheral Functions
001b	Timer
010b	Three-phase motor control timers
011b	UART
100b	UART special function
101b	Intelligent I/O groups 0 and 2, CAN channel 0
110b	Intelligent I/O group 1
111b	

Port P0_i Function Select Register (i = 0 to 7)



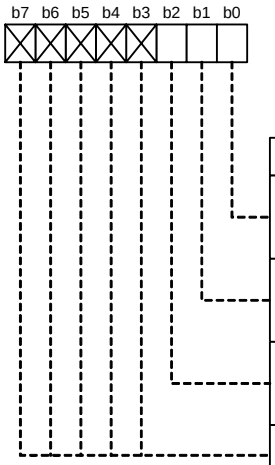
Symbol	Address	Reset Value
P0_0S to P0_2S	400A0h, 400A2h, 400A4h	0XXX X000b
P0_3S to P0_5S	400A6h, 400A8h, 400AAh	0XXX X000b
P0_6S, P0_7S	400ACh, 400AEh	0XXX X000b

Bit Symbol	Bit Name	Function	RW
PSEL0	Port P0_i Output Function Select Bit	b2 b1 b0 0 0 0 : I/O port P0_i 0 0 1 : Do not use this combination 0 1 0 : Do not use this combination 0 1 1 : Do not use this combination 1 0 0 : Do not use this combination 1 0 1 : Do not use this combination 1 1 0 : Do not use this combination 1 1 1 : Do not use this combination	RW
PSEL1			RW
PSEL2			RW
— (b6-b3)	No register bits; should be written with 0 and read as undefined value		—
ASEL	Port P0_i Analog Function Select Bit	0: Function other than AN0_i 1: AN0_i	RW

Figure 25.3 Registers P0_0S to P0_7S

The port P0_i (i = 0 to 7) shares a pin with the AN0_i input pin for the A/D converter.
To use as the programmable I/O port, the P0_iS register should be set to 00h. To use as the A/D converter input pin, this register should be set to 80h and the PD0_i bit should be set to 0 (Port P0_i functions as input port).

Port P1_i Function Select Register (i = 0 to 7)



Symbol	Address	Reset Value
P1_0S to P1_2S	400A1h, 400A3h, 400A5h	XXXX X000b
P1_3S to P1_5S	400A7h, 400A9h, 400ABh	XXXX X000b
P1_6S, P1_7S	400ADh, 400AFh	XXXX X000b

Bit Symbol	Bit Name	Function	RW
PSEL0	Port P1_i Output Function Select Bit (1)	b2 b1 b0 0 0 0 : I/O port P1_i 0 0 1 : Do not use this combination 0 1 0 : Do not use this combination 0 1 1 : Do not use this combination 1 0 0 : Do not use this combination 1 0 1 : IIO0_i output 1 1 0 : IIO1_i output 1 1 1 : Do not use this combination	RW
PSEL1			RW
PSEL2			RW
— (b7-b3)	No register bits; should be written with 0 and read as undefined value		—

Notes:
1. Refer to the table below for each pin setting.

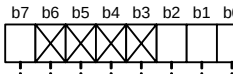
Port	Bits PSEL2 to PSEL0 Setting Value							
	000b	001b	010b	011b	100b	101b	110b	111b
P1_0	P1_0	— (2)	— (2)	— (2)	— (2)	IIO0_0 output	IIO1_0 output	— (2)
P1_1	P1_1	— (2)	— (2)	— (2)	— (2)	IIO0_1 output	IIO1_1 output	— (2)
P1_2	P1_2	— (2)	— (2)	— (2)	— (2)	IIO0_2 output	IIO1_2 output	— (2)
P1_3	P1_3	— (2)	— (2)	— (2)	— (2)	IIO0_3 output	IIO1_3 output	— (2)
P1_4	P1_4	— (2)	— (2)	— (2)	— (2)	IIO0_4 output	IIO1_4 output	— (2)
P1_5	P1_5	— (2)	— (2)	— (2)	— (2)	IIO0_5 output	IIO1_5 output	— (2)
P1_6	P1_6	— (2)	— (2)	— (2)	— (2)	IIO0_6 output	IIO1_6 output	— (2)
P1_7	P1_7	— (2)	— (2)	— (2)	— (2)	IIO0_7 output	IIO1_7 output	— (2)

2. Do not use this combination.

Figure 25.4 Registers P1_0S to P1_7S

The port P1_i (i = 0 to 7) shares a pin with the intelligent I/O groups 0 and 1 (IIO0 and IIO1), and the external interrupt input pin.
To use as an output pin, the PD1_i bit should be set to 1 (Port P1_i functions as output port) and a function should be selected according to the Figure 25.4. To use as an input pin, the PD1_i bit should be set to 0 (Port P1_i functions as input port).

Port P2_i Function Select Register (i = 0 to 7)



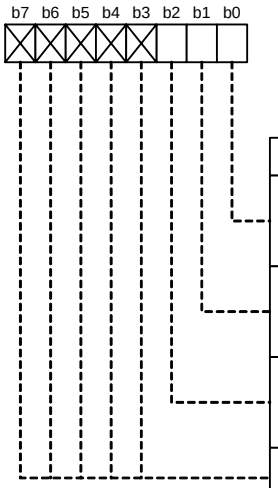
Symbol	Address	Reset Value
P2_0S to P2_2S	400B0h, 400B2h, 400B4h	0XXX X000b
P2_3S to P2_5S	400B6h, 400B8h, 400BAh	0XXX X000b
P2_6S, P2_7S	400BCh, 400BEh	0XXX X000b

Bit Symbol	Bit Name	Function	RW
PSEL0	Port P2_i Output Function Select Bit	b2 b1 b0 0 0 0 : I/O port P2_i 0 0 1 : Do not use this combination 0 1 0 : Do not use this combination 0 1 1 : Do not use this combination 1 0 0 : Do not use this combination 1 0 1 : Do not use this combination 1 1 0 : Do not use this combination 1 1 1 : Do not use this combination	RW
PSEL1			RW
PSEL2			RW
— (b6-b3)	No register bits; should be written with 0 and read as undefined value		—
ASEL	Port P2_i Analog Function Select Bit	0: Function other than AN2_i 1: AN2_i	RW

Figure 25.5 Registers P2_0S to P2_7S

The port P2_i (i = 0 to 7) shares a pin with the AN2_i pin for the A/D converter.
To use as the programmable I/O port, the P2_iS register should be set to 00h. To use as the A/D converter input pin, this register should be set to 80h and the PD2_i bit should be set to 0 (Port P2_i functions as input port).

Port P3_i Function Select Register (i = 0 to 7)



Symbol	Address	Reset Value
P3_0S to P3_2S	400B1h, 400B3h, 400B5h	XXXX X000b
P3_3S to P3_5S	400B7h, 400B9h, 400BBh	XXXX X000b
P3_6S, P3_7S	400BDh, 400BFh	XXXX X000b

Bit Symbol	Bit Name	Function	RW
PSEL0	Port P3_i Output Function Select Bit (1)	b2 b1 b0 0 0 0 : I/O port P3_i 0 0 1 : Timer output 0 1 0 : Three-phase motor control output 0 1 1 : Do not use this combination 1 0 0 : Do not use this combination 1 0 1 : Do not use this combination 1 1 0 : Do not use this combination 1 1 1 : Do not use this combination	RW
PSEL1			RW
PSEL2			RW
— (b7-b3)	No register bits; should be written with 0 and read as undefined value		—

Notes:
1. Refer to the table below for each pin setting.

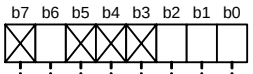
Port	Bits PSEL2 to PSEL0 Setting Value							
	000b	001b	010b	011b	100b	101b	110b	111b
P3_0	P3_0	TA0OUT output	— (2)	— (2)	— (2)	— (2)	— (2)	— (2)
P3_1	P3_1	TA3OUT output	— (2)	— (2)	— (2)	— (2)	— (2)	— (2)
P3_2	P3_2	TA1OUT output	V	— (2)	— (2)	— (2)	— (2)	— (2)
P3_3	P3_3	— (2)	$\overline{V}$	— (2)	— (2)	— (2)	— (2)	— (2)
P3_4	P3_4	TA2OUT output	W	— (2)	— (2)	— (2)	— (2)	— (2)
P3_5	P3_5	— (2)	$\overline{W}$	— (2)	— (2)	— (2)	— (2)	— (2)
P3_6	P3_6	TA4OUT output	U	— (2)	— (2)	— (2)	— (2)	— (2)
P3_7	P3_7	— (2)	$\overline{U}$	— (2)	— (2)	— (2)	— (2)	— (2)

2. Do not use this combination.

Figure 25.6 Registers P3_0S to P3_7S

The port P3_i (i = 0 to 7) shares a pin with the timer output and the three-phase motor control output. To use as an output pin, the PD3_i bit should be set to 1 (Port P3_i functions as output port) and a function should be selected according to the Figure 25.6. To use as an input pin, the PD3_i register should be set to 0 (Port P3_i functions as input port).

Port P4_i Function Select Register (i = 0 to 7)



Symbol	Address	Reset Value
P4_0S to P4_2S	400C0h, 400C2h, 400C4h	X0XX X000b
P4_3S to P4_5S	400C6h, 400C8h, 400CAh	X0XX X000b
P4_6S, P4_7S	400CCh, 400CEh	X0XX X000b

Bit Symbol	Bit Name	Function	RW
PSEL0	Port P4_i Output Function Select Bit (1)	b2 b1 b0 0 0 0 : I/O port P4_i 0 0 1 : Do not use this combination 0 1 0 : Do not use this combination 0 1 1 : UART3/UART6 output 1 0 0 : UART3/UART6 special function output 1 0 1 : IIO2 output 1 1 0 : Do not use this combination 1 1 1 : Do not use this combination	RW
PSEL1			RW
PSEL2			RW
— (b5-b3)	No register bits; should be written with 0 and read as undefined value		—
NOD	N-channel Open Drain Output Select Bit	0: Push-pull output 1: N-channel open drain output	RW
— (b7)	No register bit; should be written with 0 and read as undefined value		—

Notes:
1. Refer to the table below for each pin setting.

Port	Bits PSEL2 to PSEL0 Setting Value							
	000b	001b	010b	011b	100b	101b	110b	111b
P4_0	P4_0	— (2)	— (2)	$\overline{\text{RTS3}}$	— (2)	— (2)	— (2)	— (2)
P4_1	P4_1	— (2)	— (2)	CLK3 output	— (2)	— (2)	— (2)	— (2)
P4_2	P4_2	— (2)	— (2)	SCL3 output	STXD3	— (2)	— (2)	— (2)
P4_3	P4_3	— (2)	— (2)	TXD3 SDA3 output	— (2)	OUTC2_0 ISTXD2 IEOUT	— (2)	— (2)
P4_4	P4_4	— (2)	— (2)	$\overline{\text{RTS6}}$	— (2)	— (2)	— (2)	— (2)
P4_5	P4_5	— (2)	— (2)	CLK6 output	— (2)	— (2)	— (2)	— (2)
P4_6	P4_6	— (2)	— (2)	SCL6 output	STXD6	— (2)	— (2)	— (2)
P4_7	P4_7	— (2)	— (2)	TXD6 SDA6 output	— (2)	— (2)	— (2)	— (2)

2. Do not use this combination.

Figure 25.7 Registers P4_0S to P4_7S

The port P4_i (i = 0 to 7) shares a pin with the serial interface (UART3 and UART6) and the intelligent I/O group 2 (IIO2).
To use as an output pin, the PD4_i bit should be set to 1 (Port P4_i functions as output port) and a function should be selected according to the Figure 25.7. To use as an input pin, the PD4_i bit should be set to 0 (Port P4_i functions as input port).
Ports P4_0 to P4_7 are 5 V tolerant inputs. To use as an I/O pin with 5 V tolerant input enabled, the NOD bit should be set to 1.

Port P5_i Function Select Register (i = 0 to 7)

b7 b6 b5 b4 b3 b2 b1 b0

Symbol	Address	Reset Value
P5_0S, P5_1S	400C1h, 400C3h	XXXX X000b
P5_2S, P5_3S	400C5h, 400C7h	XXXX X000b
P5_4S, P5_5S	400C9h, 400CBh	X0XX X000b
P5_6S, P5_7S	400CDh, 400CFh	X0XX X000b

Bit Symbol	Bit Name	Function	RW
PSEL0	Port P5_i Output Function Select Bit (1)	b2 b1 b0 0 0 0 : I/O port P5_i 0 0 1 : Do not use this combination 0 1 0 : Do not use this combination 0 1 1 : UART7 output 1 0 0 : Do not use this combination 1 0 1 : Do not use this combination 1 1 0 : Do not use this combination 1 1 1 : Do not use this combination	RW
PSEL1			RW
PSEL2			RW
— (b5-b3)	No register bits; should be written with 0 and read as undefined value		—
— (b6) (i = 0 to 3) NOD (i = 4 to 7)	No register bit; should be written with 0 and read as undefined value		—
	N-channel Open Drain Output Select Bit	0: Push-pull output 1: N-channel open drain output	RW
— (b7)	No register bit; should be written with 0 and read as undefined value		—

Notes:
1. Refer to the table below for each pin setting.

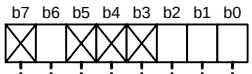
Port	Bits PSEL2 to PSEL0 Setting Value							
	000b	001b	010b	011b	100b	101b	110b	111b
P5_0	P5_0	— (2)	— (2)	— (2)	— (2)	— (2)	— (2)	— (2)
P5_1	P5_1	— (2)	— (2)	— (2)	— (2)	— (2)	— (2)	— (2)
P5_2	P5_2	— (2)	— (2)	— (2)	— (2)	— (2)	— (2)	— (2)
P5_3	P5_3	— (2)	— (2)	— (2)	— (2)	— (2)	— (2)	— (2)
P5_4	P5_4	— (2)	— (2)	TXD7	— (2)	— (2)	— (2)	— (2)
P5_5	P5_5	— (2)	— (2)	CLK7 output	— (2)	— (2)	— (2)	— (2)
P5_6	P5_6	— (2)	— (2)	— (2)	— (2)	— (2)	— (2)	— (2)
P5_7	P5_7	— (2)	— (2)	$\overline{\text{RTS7}}$	— (2)	— (2)	— (2)	— (2)

2. Do not use this combination.

Figure 25.8 Registers P5_0S to P5_7S

The port P5_i (i = 0 to 7) shares a pin with the serial interface (UART7). To use as an output pin, the PD5_i bit should be set to 1 (Port P5_i functions as output port) and a function should be selected according to the Figure 25.8. To use as an input pin, the PD5_i bit should be set to 0 (Port P5_i functions as input port).
Ports P5_4 to P5_7 are 5 V tolerant inputs. To use as an I/O pin with 5 V tolerant input enabled, the NOD bit should be set to 1.

Port P6_i Function Select Register (i = 0 to 7)



Symbol	Address	Reset Value
P6_0S to P6_2S	400D0h, 400D2h, 400D4h	X0XX X000b
P6_3S to P6_5S	400D6h, 400D8h, 400DAh	X0XX X000b
P6_6S, P6_7S	400DCh, 400DEh	X0XX X000b

Bit Symbol	Bit Name	Function	RW
PSEL0	Port P6_i Output Function Select Bit (1)	b2 b1 b0 0 0 0 : I/O port P6_i 0 0 1 : Do not use this combination 0 1 0 : Do not use this combination 0 1 1 : UART0/UART1 output 1 0 0 : UART0/UART1 special function output 1 0 1 : IIO2 output 1 1 0 : Do not use this combination 1 1 1 : Do not use this combination	RW
PSEL1			RW
PSEL2			RW
— (b5-b3)	No register bits; should be written with 0 and read as undefined value		—
NOD	N-channel Open Drain Output Select Bit	0: Push-pull output 1: N-channel open drain output	RW
— (b7)	No register bit; should be written with 0 and read as undefined value		—

Notes:
1. Refer to the table below for each pin setting.

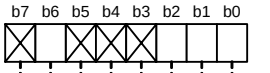
Port	Bits PSEL2 to PSEL0 Setting Value							
	000b	001b	010b	011b	100b	101b	110b	111b
P6_0	P6_0	— (2)	— (2)	$\overline{\text{RTS0}}$	— (2)	— (2)	— (2)	— (2)
P6_1	P6_1	— (2)	— (2)	CLK0 output	— (2)	— (2)	— (2)	— (2)
P6_2	P6_2	— (2)	— (2)	SCL0 output	STXD0	— (2)	— (2)	— (2)
P6_3	P6_3	— (2)	— (2)	TXD0 SDA0 output	— (2)	— (2)	— (2)	— (2)
P6_4	P6_4	— (2)	— (2)	$\overline{\text{RTS1}}$	— (2)	OUTC_1 ISCLK2 output	— (2)	— (2)
P6_5	P6_5	— (2)	— (2)	CLK1 output	— (2)	— (2)	— (2)	— (2)
P6_6	P6_6	— (2)	— (2)	SCL1 output	STXD1	— (2)	— (2)	— (2)
P6_7	P6_7	— (2)	— (2)	TXD1 SDA1 output	— (2)	— (2)	— (2)	— (2)

2. Do not use this combination.

Figure 25.9 Registers P6_0S to P6_7S

The port P6_i (i = 0 to 7) shares a pin with the serial interface (UART0 and UART1) and the intelligent I/O group 2 (IIO2).
To use as an output pin, the PD6_i bit should be set to 1 (Port P6_i functions as output port) and a function should be selected according to the Figure 25.9. To use as an input pin, the PD6_i bit should be set to 0 (Port P6_i functions as input port).
Ports P6_0 to P6_7 are 5 V tolerant inputs. To use as an I/O pin with 5 V tolerant input enabled, the NOD bit should be set to 1.

Port P7_i Function Select Register (i = 0 to 7)



Symbol	Address	Reset Value
P7_0S to P7_2S	400D1h, 400D3h, 400D5h	X0XX X000b
P7_3S to P7_5S	400D7h, 400D9h, 400DBh	X0XX X000b
P7_6S, P7_7S	400DDh, 400DFh	X0XX X000b

Bit Symbol	Bit Name	Function	RW
PSEL0	Port P7_i Output Function Select Bit (1)	b2 b1b0 0 0 0 : I/O port P7_i 0 0 1 : Timer output 0 1 0 : Three-phase motor control output 0 1 1 : UART2/UART5/MMI2C output 1 0 0 : UART2 special function output 1 0 1 : IIO2/CAN0 output 1 1 0 : IIO1 output 1 1 1 : Do not use this combination	RW
PSEL1			RW
PSEL2			RW
— (b5-b3)	No register bits; should be written with 0 and read as undefined value		—
NOD	N-channel Open Drain Output Select Bit	0: Push-pull output 1: N-channel open drain output	RW
— (b7)	No register bit; should be written with 0 and read as undefined value		—

Notes:
1. Refer to the table below for each pin setting.

Port	Bits PSEL2 to PSEL0 Setting Value							
	000b	001b	010b	011b	100b	101b	110b	111b
P7_0	P7_0	TA0OUT output	— (2)	TXD2 SDA2 output MSDA output	— (2)	OUTC2_0 ISTXD2 IEOUT	IIO1_6 output	— (2)
P7_1	P7_1	— (2)	— (2)	SCL2 output MSCL output	STXD2	OUTC2_2	IIO1_7 output	— (2)
P7_2	P7_2	TA1OUT output	V	CLK2 output	— (2)	— (2)	— (2)	— (2)
P7_3	P7_3	— (2)	$\overline{V}$	$\overline{RTS2}$	— (2)	— (2)	IIO1_0 o—Í	— (2)
P7_4	P7_4	TA2OUT output	W	— (2)	— (2)	— (2)	IIO1_1 output	— (2)
P7_5	P7_5	— (2)	$\overline{W}$	— (2)	— (2)	— (2)	IIO1_2 output	— (2)
P7_6	P7_6	TA3OUT output	— (2)	TXD5 SDA5 output	— (2)	CAN0OUT	IIO1_3 output	— (2)
P7_7	P7_7	— (2)	— (2)	CLK5 output	— (2)	— (2)	IIO1_4 output	— (2)

2. Do not use this combination.

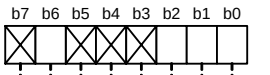
Figure 25.10 Registers P7_0S to P7_7S

The port P7_i (i=0 to 7) shares a pin with the timer, the three-phase motor control, the serial interface (UART2 and UART5), the multi-master I^2C-bus interface (MMI2C), the intelligent I/O groups 1 and 2 (IIO1 and IIO2), and the CAN module.

To use as an output pin, the PD7_i bit should be set to 1 (Port P7_i functions as output port) and a function should be selected according to the Figure 25.10. To use as an input pin, the PD7_i bit should be set to 0 (Port P7_i functions as input port).

Ports P7_0 to P7_7 are 5 V tolerant inputs. To use as an I/O pin with 5 V tolerant input enabled, the NOD bit should be set to 1.

Port P8_i Function Select Register (i = 0 to 4, 6, 7)



Symbol	Address	Reset Value
P8_0S, P8_1S	400E0h, 400E2h	X0XX X000b
P8_2S, P8_3S	400E4h, 400E6h	X0XX X000b
P8_4S	400E8h	XXXX X000b
P8_6S, P8_7S	400ECh, 400EEh	XXXX X000b

Bit Symbol	Bit Name	Function	RW
PSEL0	Port P8_i Output Function Select Bit (1)	b2 b1 b0 0 0 0 : I/O port P8_i 0 0 1 : Timer output 0 1 0 : Three-phase motor control output 0 1 1 : UART5 output 1 0 0 : UART5 special function output 1 0 1 : CAN0 output 1 1 0 : IIO1 output 1 1 1 : Do not use this combination	RW
PSEL1			RW
PSEL2			RW
— (b5-b3)	No register bits; should be written with 0 and read as undefined value		—
NOD (i = 0 to 3) — (b6) (i = 4, 6, 7)	N-channel Open Drain Output Select Bit	0: Push-pull output 1: N-channel open drain output	RW
	No register bit; should be written with 0 and read as undefined value		—
— (b7)	No register bit; should be written with 0 and read as undefined value		—

Notes:
1. Refer to the table below for each pin setting.

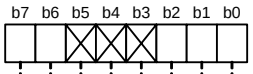
Port	Bits PSEL2 to PSEL0 Setting Value							
	000b	001b	010b	011b	100b	101b	110b	111b
P8_0	P8_0	TA4OUT output	U	SCL5 output	STXD5	— (2)	— (2)	— (2)
P8_1	P8_1	— (2)	$\overline{U}$	$\overline{RTS5}$	— (2)	— (2)	IIO1_5 output	— (2)
P8_2	P8_2	— (2)	— (2)	— (2)	— (2)	CAN0OUT	— (2)	— (2)
P8_3	P8_3	— (2)	— (2)	— (2)	— (2)	— (2)	— (2)	— (2)
P8_4	P8_4	— (2)	— (2)	— (2)	— (2)	— (2)	— (2)	— (2)
P8_6	P8_6	— (2)	— (2)	— (2)	— (2)	— (2)	— (2)	— (2)
P8_7	P8_7	— (2)	— (2)	— (2)	— (2)	— (2)	— (2)	— (2)

2. Do not use this combination.

Figure 25.11 Registers P8_0S to P8_4S, P8_6S, and P8_7S

The port P8_i (i = 0 to 4, 6, and 7) shares a pin with the timer, the three-phase motor control, the serial interface (UART5), the intelligent I/O group 1 (IIO1), the CAN module, and the external interrupt input pin. To use as an output pin, the PD8_i bit should be set to 1 (Port P8_i functions as output port) and a function should be selected according to the Figure 25.11. To use as an input pin, the PD8_i bit should be set to 0 (Port P8_i functions as input port).
Ports P8_0 to P8_3 are 5 V tolerant inputs. To use as an I/O pin with 5 V tolerant input enabled, the NOD bit should be set to 1.

Port P9_i Function Select Register (i = 0 to 7) (1)



Symbol	Address	Reset Value
P9_0S to P9_2S	400E1h, 400E3h, 400E5h	X0XX X000b
P9_3S to P9_5S	400E7h, 400E9h, 400EBh	00XX X000b
P9_6S	400EDh	00XX X000b
P9_7S	400EFh	X0XX X000b

Bit Symbol	Bit Name	Function	RW
PSEL0	Port P9_i Output Function Select Bit (2)	b2 b1 b0 0 0 0 : I/O port P9_i 0 0 1 : Do not use this combination 0 1 0 : Do not use this combination 0 1 1 : UART3/UART4 output 1 0 0 : UART3/UART4 special function output 1 0 1 : IIO2 output 1 1 0 : Do not use this combination 1 1 1 : Do not use this combination	RW
PSEL1			RW
PSEL2			RW
— (b5-b3)	No register bits; should be written with 0 and read as undefined value		—
NOD	N-channel Open Drain Output Select Bit	0: Push-pull output 1: N-channel open drain output	RW
— (b7) (i = 0 to 2, 7)	No register bit; should be written with 0 and read as undefined value		—
ASEL (i = 3 to 6)	Port P9_i (i = 3 to 6) Analog Functions Select Bit	0: Function other than Analog pin 1: Analog pin	RW

Notes:
1. The instruction to set this register should be written immediately after the instruction to set the PRC2 bit in the PRCR register to 1 (write enabled). Any interrupt or DMA transfer should not be generated between these two instructions.
2. Refer to the table below for each pin setting.

Port	Bits PSEL2 to PSEL0 Setting Value							
	000b	001b	010b	011b	100b	101b	110b	111b
P9_0	P9_0	— (3)	— (3)	CLK3 output	— (3)	— (3)	— (3)	— (3)
P9_1	P9_1	— (3)	— (3)	SCL3 output	STXD3	— (3)	— (3)	— (3)
P9_2	P9_2	— (3)	— (3)	TXD3 SDA3 output	— (3)	OUTC2_0 ISTXD2 IEOUT	— (3)	— (3)
P9_3	P9_3	— (3)	— (3)	$\overline{\text{RTS3}}$	— (3)	— (3)	— (3)	— (3)
P9_4	P9_4	— (3)	— (3)	$\overline{\text{RTS4}}$	— (3)	— (3)	— (3)	— (3)
P9_5	P9_5	— (3)	— (3)	CLK4 output	— (3)	— (3)	— (3)	— (3)
P9_6	P9_6	— (3)	— (3)	TXD4 SDA4 output	— (3)	— (3)	— (3)	— (3)
P9_7	P9_7	— (3)	— (3)	SCL4 output	STXD4	— (3)	— (3)	— (3)

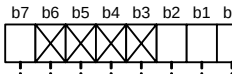
3. Do not use this combination.

Figure 25.12 Registers P9_0S to P9_7S

The port P9_i (i = 0 to 7) shares a pin with the serial interface (UART3 and UART4) and the intelligent I/O group 2 (IIO2). In particular, the port P9_i (i = 3 to 6) also shares a pin with the A/D converter I/O (ANEX0 and ANEX1) pin and the D/A converter output pin.
To use as the A/D converter pin or the D/A converter pin, the P9_iS register should be set to 80h and the PD9_i bit should be set to 0 (Port P9_i functions as input port) irrespective of the input/output state.

To use as an output pin of functions other than the A/D converter or the D/A converter, the PD9_i bit should be set to 1 (Port P9_i functions as output port) and a function should be selected according to the Figure 25.12. To use as an input pin of functions other than the A/D converter or the D/A converter, the PD9_i bit should be set to 0 (Port P9_i functions as input port).
When the NOD bit is set to 1, the corresponding pin functions as an N-channel open drain output.

Port P10_i Function Select Register (i = 0 to 7)



Symbol	Address	Reset Value
P10_0S to P10_2S	400F0h, 400F2h, 400F4h	0XXX X000b
P10_3S to P10_5S	400F6h, 400F8h, 400FAh	0XXX X000b
P10_6S, P10_7S	400FCh, 400FEh	0XXX X000b

Bit Symbol	Bit Name	Function	RW
PSEL0	Port P10_i Output Function Select Bit	b2 b1 b0 0 0 0 : I/O port P10_i 0 0 1 : Do not use this combination 0 1 0 : Do not use this combination 0 1 1 : Do not use this combination 1 0 0 : Do not use this combination 1 0 1 : Do not use this combination 1 1 0 : Do not use this combination 1 1 1 : Do not use this combination	RW
PSEL1			RW
PSEL2			RW
— (b6-b3)	No register bits; should be written with 0 and read as undefined value		—
ASEL	Port P10_i Analog Functions Select Bit	0: Function other than AN_i 1: AN_i	RW

Figure 25.13 Registers P10_0S to P10_7S

The port P10_i (i = 0 to 7) shares a pin with the AN_i input pin for the A/D converter and the key input interrupt pin.

To use as the programmable I/O port, the P10_iS register should be set to 00h. To use as an input pin (except for the A/D converter), the PD10_i bit should be set to 0 (Port P10_i functions as input port). To use as an input pin for the A/D converter, the P10_iS register should be set to 80h and the PD10_i bit should be set to 0 (Port P10_i functions as input port).

Port P11_i Function Select Register (i = 0 to 4)

b7 b6 b5 b4 b3 b2 b1 b0

Symbol	Address	Reset Value
P11_0S to P11_2S	400F1h, 400F3h, 400F5h	X0XX X000b
P11_3S	400F7h	X0XX X000b
P11_4S	400F9h	XXXX X000b

Bit Symbol	Bit Name	Function	RW
PSEL0	Port P11_i Output Function Select Bit (1)	b2 b1 b0 0 0 0 : I/O port P11_i 0 0 1 : Do not use this combination 0 1 0 : Do not use this combination 0 1 1 : UART8 output 1 0 0 : Do not use this combination 1 0 1 : Do not use this combination 1 1 0 : IIO1 output 1 1 1 : Do not use this combination	RW
PSEL1			RW
PSEL2			RW
— (b5-b3)	No register bits; should be written with 0 and read as undefined value		—
NOD (i = 0 to 3)	N-channel Open Drain Output Select Bit	0: Push-pull output 1: N-channel open drain output	RW
— (b6) (i = 4)	No register bit; should be written with 0 and read as undefined value		—
— (b7)	No register bit; should be written with 0 and read as undefined value		—

Notes:
1. Refer to the table below for each pin setting.

Port	Bits PSEL2 to PSEL0 Setting Value							
	000b	001b	010b	011b	100b	101b	110b	111b
P11_0	P11_0	— (2)	— (2)	TXD8	— (2)	— (2)	— (2)	— (2)
P11_1	P11_1	— (2)	— (2)	CLK8 output	— (2)	— (2)	— (2)	— (2)
P11_2	P11_2	— (2)	— (2)	— (2)	— (2)	— (2)	— (2)	— (2)
P11_3	P11_3	— (2)	— (2)	— (2)	— (2)	— (2)	IIO1_3 output	— (2)
P11_4	P11_4	— (2)	— (2)	— (2)	— (2)	— (2)	— (2)	— (2)

2. Do not use this combination.

Figure 25.14 Registers P11_0S to P11_4S

The port P11_i (i = 0 to 4) shares a pin with the serial interface (UART8) and the intelligent I/O group 1 (IIO1).
Serial interface (UART8) is used for the communication with audio DSP. Set registers P11_0S and P11_1S to 03h (UART8 output), and the P11_2S register to 00h (I/O port P11_i).
To use ports P11_3 and P11_4 as an output pins, the PD11_i bit should be set to 1 (Port P11_i functions as output port) and a function should be selected according to the Figure 25.14. To use as an input pin, the PD11_i bit should be set to 0 (Port P11_i functions as input port).
To use as an N-channel open drain output, the NOD bit should be set to 1.

Port P12_i Function Select Register (i = 0 to 7)

b7 b6 b5 b4 b3 b2 b1 b0

Symbol	Address	Reset Value
P12_0S to P12_2S	40100h, 40102h, 40104h	X0XX X000b
P12_3S	40106h	X0XX X000b
P12_4S to P12_6S	40108h, 4010Ah 4010Ch	XXXX X000b
P12_7S	4010Eh	XXXX X000b

Bit Symbol	Bit Name	Function	RW
PSEL0	Port P12_i Output Function Select Bit (1)	b2 b1 b0 0 0 0 : I/O port P12_i 0 0 1 : Do not use this combination 0 1 0 : Do not use this combination 0 1 1 : UART6 output 1 0 0 : UART6 special function 1 0 1 : Do not use this combination 1 1 0 : Do not use this combination 1 1 1 : Do not use this combination	RW
PSEL1			RW
PSEL2			RW
— (b5-b3)	No register bits; should be written with 0 and read as undefined value		—
NOD (i = 0 to 3)	N-channel Open Drain Output Select Bit	0: Push-pull output 1: N-channel open drain output	RW
— (b6) (i = 4 to 7)	No register bits; should be written with 0 and read as undefined value		—
— (b7)	No register bit; should be written with 0 and read as undefined value		—

Notes:
1. Refer to the table below for each pin setting.

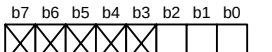
Port	Bits PSEL2 to PSEL0 Setting Value							
	000b	001b	010b	011b	100b	101b	110b	111b
P12_0	P12_0	— (2)	— (2)	TXD6 SDA6 output	— (2)	— (2)	— (2)	— (2)
P12_1	P12_1	— (2)	— (2)	CLK6 output	— (2)	— (2)	— (2)	— (2)
P12_2	P12_2	— (2)	— (2)	SCL6 output	STXD6	— (2)	— (2)	— (2)
P12_3	P12_3	— (2)	— (2)	$\overline{\text{RTS6}}$	— (2)	— (2)	— (2)	— (2)
P12_4	P12_4	— (2)	— (2)	— (2)	— (2)	— (2)	— (2)	— (2)
P12_5	P12_5	— (2)	— (2)	— (2)	— (2)	— (2)	— (2)	— (2)
P12_6	P12_6	— (2)	— (2)	— (2)	— (2)	— (2)	— (2)	— (2)
P12_7	P12_7	— (2)	— (2)	— (2)	— (2)	— (2)	— (2)	— (2)

2. Do not use this combination.

Figure 25.15 Registers P12_0S to P12_7S

The port P12_i (i = 0 to 7) shares a pin with the serial interface (UART6).
To use as an output pin, the PD12_i bit should be set to 1 (Port P12_i functions as output port) and a function should be selected according to the Figure 25.15. To use as an input pin, the PD12_i bit should be set to 0 (Port P12_i functions as input port).
When the NOD bit is set to 1, the corresponding pin functions as an N-channel open drain output.

Port P13_i Function Select Register (i = 0 to 7)



Symbol	Address	Reset Value
P13_0S to P13_2S	40101h, 40103h, 40105h	XXXX X000b
P13_3S to P13_5S	40107h, 40109h, 4010Bh	XXXX X000b
P13_6S, P13_7S	4010Dh, 4010Fh	XXXX X000b

Bit Symbol	Bit Name	Function	RW
PSEL0	Port P13_i Output Function Select Bit (1)	b2 b1 b0 0 0 0 : I/O port P13_i 0 0 1 : Do not use this combination 0 1 0 : Do not use this combination 0 1 1 : Do not use this combination 1 0 0 : Do not use this combination 1 0 1 : IIO2 output 1 1 0 : Do not use this combination 1 1 1 : Do not use this combination	RW
PSEL1			RW
PSEL2			RW
— (b7-b3)	No register bits; should be written with 0 and read as undefined value		—

Notes:
1. Refer to the table below for each pin setting.

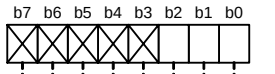
Port	Bits PSEL2 to PSEL0 Setting Value							
	000b	001b	010b	011b	100b	101b	110b	111b
P13_0	P13_0	— (2)	— (2)	— (2)	— (2)	OUTC2_4	— (2)	— (2)
P13_1	P13_1	— (2)	— (2)	— (2)	— (2)	OUTC2_5	— (2)	— (2)
P13_2	P13_2	— (2)	— (2)	— (2)	— (2)	OUTC2_6	— (2)	— (2)
P13_3	P13_3	— (2)	— (2)	— (2)	— (2)	OUTC2_3	— (2)	— (2)
P13_4	P13_4	— (2)	— (2)	— (2)	— (2)	OUTC2_0 ISTXD2 IEOUT	— (2)	— (2)
P13_5	P13_5	— (2)	— (2)	— (2)	— (2)	OUTC2_2	— (2)	— (2)
P13_6	P13_6	— (2)	— (2)	— (2)	— (2)	OUTC2_1 ISCLK2 output	— (2)	— (2)
P13_7	P13_7	— (2)	— (2)	— (2)	— (2)	OUTC2_7	— (2)	— (2)

2. Do not use this combination.

Figure 25.16 Registers P13_0S to P13_7S

The port P13_i (i = 0 to 7) shares a pin with the intelligent I/O group 2 (IIO2).
To use as an output pin, the PD13_i bit should be set to 1 (Port P13_i functions as output port) and a function should be selected according to the Figure 25.16. To use as an input pin, the PD13_i bit should be set to 0 (Port P13_i functions as input port).

Port P14_i Function Select Register (i = 3 to 6)



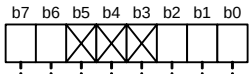
Symbol	Address	Reset Value
P14_3S to P14_5S	40116h, 40118h, 4011Ah	XXXX X000b
P14_6S	4011Ch	XXXX X000b

Bit Symbol	Bit Name	Function	RW
PSEL0	Port P14_i Output Function Select Bit	b2 b1 b0 0 0 0 : I/O port P14_i 0 0 1 : Do not use this combination 0 1 0 : Do not use this combination 0 1 1 : Do not use this combination 1 0 0 : Do not use this combination 1 0 1 : Do not use this combination 1 1 0 : Do not use this combination 1 1 1 : Do not use this combination	RW
PSEL1			RW
PSEL2			RW
— (b7-b3)	No register bits; should be written with 0 and read as undefined value		—

Figure 25.17 Registers P14_3S to P14_6S

The port P14_i (i = 3 to 6) shares a pin with the external interrupt input pin. The P14_iS register should be set to 00h (I/O port).

Port P15_i Function Select Register (i = 0 to 7)



Symbol	Address	Reset Value
P15_0S to P15_2S	40111h, 40113h, 40115h	00XX X000b
P15_3S to P15_5S	40117h, 40119h, 4011Bh	00XX X000b
P15_6S, P15_7S	4011Dh, 4011Fh	00XX X000b

Bit Symbol	Bit Name	Function	RW
PSEL0	Port P15_i Output Function Select Bit (1)	b2 b1 b0 0 0 0 : I/O port P15_i 0 0 1 : Do not use this combination 0 1 0 : Do not use this combination 0 1 1 : UART6/UART7 output 1 0 0 : UART6 special function 1 0 1 : IIO0_i output 1 1 0 : Do not use this combination 1 1 1 : Do not use this combination	RW
PSEL1			RW
PSEL2			RW
— (b5-b3)	No register bits; should be written with 0 and read as undefined value		—
NOD	N-channel Open Drain Output Select Bit	0: Push-pull output 1: N-channel open drain output	RW
ASEL	Port P15_i Analog Function Select Bit	0: Function other than AN15_i 1: AN15_i	RW

Notes:
1. Refer to the table below for each pin setting.

Port	Bits PSEL2 to PSEL0 Setting Value							
	000b	001b	010b	011b	100b	101b	110b	111b
P15_0	P15_0	— (2)	— (2)	TXD7	— (2)	IIO0_0 output	— (2)	— (2)
P15_1	P15_1	— (2)	— (2)	CLK7 output	— (2)	IIO0_1 output	— (2)	— (2)
P15_2	P15_2	— (2)	— (2)	— (2)	— (2)	IIO0_2 output	— (2)	— (2)
P15_3	P15_3	— (2)	— (2)	$\overline{\text{RTS7}}$	— (2)	IIO0_3 output	— (2)	— (2)
P15_4	P15_4	— (2)	— (2)	TXD6 SDA6 output	— (2)	IIO0_4 output	— (2)	— (2)
P15_5	P15_5	— (2)	— (2)	SCL6 output	STXD6	IIO0_5 output	— (2)	— (2)
P15_6	P15_6	— (2)	— (2)	CLK6 output	— (2)	IIO0_6 output	— (2)	— (2)
P15_7	P15_7	— (2)	— (2)	$\overline{\text{RTS6}}$	— (2)	IIO0_7 output	— (2)	— (2)

2. Do not use this combination.

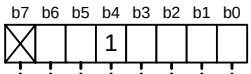
Figure 25.18 Registers P15_0S to P15_7S

The port P15_i (i = 0 to 7) shares a pin with the serial interface (UART6 and UART7), the intelligent I/O group 0 (IIO0), and the AN15_i input pin for the A/D converter.
Ports P15_4 to P15_7 are used for the control of audio CODEC. Set registers P15_4S and P15_6S to 00h (I/O port P15_i) or 03h (UART6), and registers P15_5S and P15_7S to 00h (I/O port P15_i).
To use ports P15_0 to P15_3 as an output pins, the PD15_i bit should be set to 1 (Port P15_i functions as output port) and a function should be selected according to the Figure 25.18. To use as an input pin (except for the A/D converter), the PD15_i bit should be set to 0 (Port P15_i functions as input port). To use as an input pin for the A/D converter, the P15_iS register should be set to 80h and the PD15_i bit should be set to 0.
To use as an N-channel open drain output, the NOD bit should be set to 1.

25.3 Input Function Select Register

When a peripheral function input is assigned to multiple pins, this register selects which input pin should be connected to the peripheral function.
Figure 25.19 to Figure 25.22 show the input function select registers.

Input Function Select Register 0



Symbol	Address	Reset Value
IFS0	40098h	X000 0000b

Bit Symbol	Bit Name	Function	RW
IFS00	Timer A Input Pin Switch Bit (1)	Assign timer A input to 0: Port P3 1: Port P7/port P8	RW
IFS01	Timer B Input Pin Switch Bit (2)	Assign timer B input to 0: Port P6 1: Port P9	RW
IFS02	UART6 Input Pin Switch Bit (3)	Assign UART6 input to b3 b2 0 0 : Port P4 0 1 : Do not use this combination 1 0 : Port P15 1 1 : Port P12	RW
IFS03			RW
IFS04	Reserved	Should be written with 1	RW
IFS05	UART7 Input Pin Switch Bit (4)	Assign UART7 input to 0: Port P5 1: Port P15	RW
IFS06	UART3 Input Pin Switch Bit (5)	Assign UART3 input to 0: Port P4 1: Port P9	RW
— (b7)	No register bit; should be written with 0 and read as undefined value		—

Notes:

1. Refer to the table below for each pin setting of the timer A.

IFS00	TA0OUT input	TA1OUT input	TA1IN	TA2OUT input	TA2IN	TA3OUT input	TA4OUT input	TA4IN
0	P3_0	P3_2	P3_3	P3_4	P3_5	P3_1	P3_6	P3_7
1	P7_0	P7_2	P7_3	P7_4	P7_5	P7_6	P8_0	P8_1

2. Refer to the table below for each pin setting of the timer B.

IFS01	TB0IN	TB1IN	TB2IN
0	P6_0	P6_1	P6_2
1	P9_0	P9_1	P9_2

3. Refer to the table below for each pin setting of UART6.

IFS03	IFS02	SDA6 input/SRXD6	RXD6/SCL6 input	CLK6 input	$\overline{\text{CTS6}}/\overline{\text{SS6}}$
0	0	P4_7	P4_6	P4_5	P4_4
1	0	P15_4	P15_5	P15_6	P15_7
1	1	P12_0	P12_2	P12_1	P12_3

4. Refer to the table below for each pin setting of UART7.

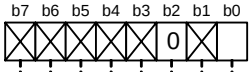
IFS05	CLK7 input	RXD7	$\overline{\text{CTS7}}$
0	P5_5	P5_6	P5_7
1	P15_1	P15_2	P15_3

5. Refer to the table below for each pin setting of UART3.

IFS06	SDA3 input/SRXD3	RXD3/SCL3 input	CLK3 input	$\overline{\text{CTS3}}/\overline{\text{SS3}}$
0	P4_3	P4_2	P4_1	P4_0
1	P9_2	P9_1	P9_0	P9_3

Figure 25.19 IFS0 Register

Input Function Select Register 1



Symbol	Address	Reset Value
IFS1	40099h	XXXX X0X0b

Bit Symbol	Bit Name	Function	RW
IFS10	CAN0 Input Pin Switch Bit	Assign CAN0IN/$\overline{\text{CAN0WU}}$ input to 0: Port P7_7 1: Port P8_3	RW
— (b1)	No register bit; should be written with 0 and read as undefined value		—
— (b2)	Reserved	Should be written with 0	RW
— (b7-b3)	No register bits; should be written with 0 and read as undefined value		—

Figure 25.20 IFS1 Register

Input Function Select Register 2

b7 b6 b5 b4 b3 b2 b1 b0

Symbol	Address	Reset Value
IFS2	4009Ah	0000 00X0b

Bit Symbol	Bit Name	Function	RW
IFS20	Intelligent I/O Group 0 Input Pin Switch Bit (1)	Assign IIO0 input to 0: Port P1 1: Port P15	RW
— (b1)	No register bit; should be written with 0 and read as undefined value		—
IFS22	Intelligent I/O Group 0 Two-phase Pulse Input Pin Switch Bit (2)	Assign this input to b3 b2 0 0 : Port P8 and $\overline{INT1}$ 0 1 : Port P7 and $\overline{INT0}$ 1 0 : Port P3 and $\overline{INT1}$ 1 1 : Port P3 and $\overline{INT0}$	RW
IFS23			RW
IFS24	Intelligent I/O Group 1 Input Pin Switch Bit (3)	Assign IIO1 input to b5 b4 0 0 : Port P7/port P8 0 1 : Port P11 1 0 : Port P1 1 1 : Do not use this combination	RW
IFS25			RW
IFS26	Intelligent I/O Group 1 Two-phase Pulse Input Pin Switch Bit (4)	Assign this input to b7 b6 0 0 : Port P8 and $\overline{INT1}$ 0 1 : Port P7 and $\overline{INT0}$ 1 0 : Port P3 and $\overline{INT1}$ 1 1 : Port P3 and $\overline{INT0}$	RW
IFS27			RW

Notes:

1. Refer to the table below for each pin setting of the Intelligent I/O group 0.

IFS20	IIO0_0 input	IIO0_1 input	IIO0_2 input	IIO0_3 input	IIO0_4 input	IIO0_5 input	IIO0_6 input	IIO0_7 input
0	P1_0	P1_1	P1_2	P1_3	P1_4	P1_5	P1_6	P1_7
1	P15_0	P15_1	P15_2	P15_3	—	—	—	—

2. Refer to the table below for each pin setting of the Intelligent I/O group 0 in two-phase pulse signal processing mode.

IFS23	IFS22	UD0A	UD0B	UD0Z
0	0	P8_0	P8_1	P8_3 ($\overline{INT1}$)
0	1	P7_6	P7_7	P8_2 ($\overline{INT0}$)
1	0	P3_0	P3_1	P8_3 ($\overline{INT1}$)
1	1	P3_0	P3_1	P8_2 ($\overline{INT0}$)

3. Refer to the table below for each pin setting of the Intelligent I/O group 1.

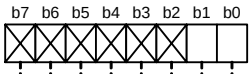
IFS25	IFS24	IIO1_0 input	IIO1_1 input	IIO1_2 input	IIO1_3 input	IIO1_4 input	IIO1_5 input	IIO1_6 input	IIO1_7 input
0	0	P7_3	P7_4	P7_5	P7_6	P7_7	P8_1	P7_0	P7_1
0	1	—	—	—	P11_3	—	—	—	—
1	0	P1_0	P1_1	P1_2	P1_3	P1_4	P1_5	P1_6	P1_7

4. Refer to the table below for each pin setting of the Intelligent I/O group 1 in two-phase pulse signal processing mode.

IFS27	IFS26	UD1A	UD1B	UD1Z
0	0	P8_0	P8_1	P8_3 ($\overline{INT1}$)
0	1	P7_6	P7_7	P8_2 ($\overline{INT0}$)
1	0	P3_0	P3_1	P8_3 ($\overline{INT1}$)
1	1	P3_0	P3_1	P8_2 ($\overline{INT0}$)

Figure 25.21 IFS2 Register

Input Function Select Register 3



Symbol	Address	Reset Value
IFS3	4009Bh	XXXX XX00b

Bit Symbol	Bit Name	Function	RW
IFS30	Intelligent I/O Group 2 Input Pin Switch Bit (1)	Assign IIO2 input to b1 b0 0 0 : Port P6/port P7 0 1 : Port P6/port P9 1 0 : Port P13 1 1 : Port P6/port P4	RW
IFS31			RW
— (b7-b2)	No register bits; should be written with 0 and read as undefined value		—

Note:
1. Refer to the table below for each pin setting of the Intelligent I/O group 2.

IFS31	IFS30	ISCLK2 input	ISRXD2/IEIN
0	0	P6_4	P7_1
0	1	P6_4	P9_1
1	0	P13_6	P13_5
1	1	P6_4	P4_2

Figure 25.22 IFS3 Register

25.4 Pull-up Control Registers 0 to 4 (Registers PUR0 to PUR4)

Figure 25.23 to Figure 25.27 show registers PUR0 to PUR4.

These registers enable/disable the pull-up resistors for every group of four pins. To enable the pull-up resistors, the corresponding bits in registers PUR0 to PUR4 should be set to 1 (pull-up resistor enabled) and the respective bits in the direction register should be set to 0 (input).

Pull-up Control Register 0

b7 b6 b5 b4 b3 b2 b1 b0

Symbol	Address	Reset Value
PUR0	03F0h	0000 0000b

Bit Symbol	Bit Name	Function	RW
PU00	P0_0 to P0_3 Pull-up Control Bit	Control pull-up setting for corresponding ports 0: Pull-up resistor disabled 1: Pull-up resistor enabled	RW
PU01	P0_4 to P0_7 Pull-up Control Bit		RW
PU02	P1_0 to P1_3 Pull-up Control Bit		RW
PU03	P1_4 to P1_7 Pull-up Control Bit		RW
PU04	P2_0 to P2_3 Pull-up Control Bit		RW
PU05	P2_4 to P2_7 Pull-up Control Bit		RW
PU06	P3_0 to P3_3 Pull-up Control Bit		RW
PU07	P3_4 to P3_7 Pull-up Control Bit		RW

Figure 25.23 PUR0 Register

Pull-up Control Register 1

b7 b6 b5 b4 b3 b2 b1 b0

Symbol	Address	Reset Value
PUR1	03F1h	XXXX X0XXb

Bit Symbol	Bit Name	Function	RW
— (b1-b0)	No register bits; should be written with 0 and read as undefined value		—
PU12	P5_0 to P5_3 Pull-up Control Bit	Control pull-up setting for corresponding ports 0: Pull-up resistor disabled 1: Pull-up resistor enabled	RW
— (b7-b3)	No register bits; should be written with 0 and read as undefined value		—

Figure 25.24 PUR1 Register

Pull-up Control Register 2

b7 b6 b5 b4 b3 b2 b1 b0

Symbol: PUR2 | Address: 03F2h | Reset Value: 000X XXXXb

Bit Symbol	Bit Name	Function	RW
— (b4-b0)	No register bits; should be written with 0 and read as undefined value		—
PU25	P8_4 to P8_7 Pull-up Control Bit (1)	Control pull-up setting for corresponding ports 0: Pull-up resistor disabled 1: Pull-up resistor enabled	RW
PU26	P9_0 to P9_3 Pull-up Control Bit		RW
PU27	P9_4 to P9_7 Pull-up Control Bit		RW

Note:
1. The port P8_5 has no pull-up resistor.

Figure 25.25 PUR2 Register

Pull-up Control Register 3

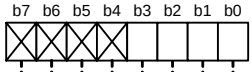
b7 b6 b5 b4 b3 b2 b1 b0

Symbol: PUR3 | Address: 03F3h | Reset Value: 0000 0000b

Bit Symbol	Bit Name	Function	RW
PU30	P10_0 to P10_3 Pull-up Control Bit	Control pull-up setting for corresponding ports 0: Pull-up resistor disabled 1: Pull-up resistor enabled	RW
PU31	P10_4 to P10_7 Pull-up Control Bit		RW
PU32	P11_0 to P11_3 Pull-up Control Bit		RW
PU33	P11_4 Pull-up Control Bit		RW
PU34	P12_0 to P12_3 Pull-up Control Bit		RW
PU35	P12_4 to P12_7 Pull-up Control Bit		RW
PU36	P13_0 to P13_3 Pull-up Control Bit		RW
PU37	P13_4 to P13_7 Pull-up Control Bit		RW

Figure 25.26 PUR3 Register

Pull-up Control Register 4



Symbol	Address	Reset Value
PUR4	03F4h	XXXX 0000b

Bit Symbol	Bit Name	Function	RW
PU40	P14_1 and P14_3 Pull-up Control Bit	Control pull-up setting for corresponding ports 0: Pull-up resistor disabled 1: Pull-up resistor enabled	RW
PU41	P14_4 to P14_6 Pull-up Control Bit		RW
PU42	P15_0 to P15_3 Pull-up Control Bit		RW
PU43	P15_4 to P15_7 Pull-up Control Bit		RW
— (b7-b4)	No register bits; should be written with 0 and read as undefined value		—

Figure 25.27 PUR4 Register

25.5 Port Control Register (PCR Register)

Figure 25.28 shows the PCR register.

This register selects an output mode for the port P1 between push-pull output and pseudo-N-channel open drain output. When the PCR0 bit is set to 1, the P-channel transistor in the output buffer is turned off. Note that the port P1 cannot be a perfect open drain output due to remaining parasitic diode. The absolute maximum rating of the input voltage is, therefore, from -0.3 V to VCC + 0.3 V (Refer to Figure 25.29).

Port Control Register

b7	b6	b5	b4	b3	b2	b1	b0
1							

Symbol: PCR
Address: 03FFh
Reset Value: 1XXX XXX0b

Bit Symbol	Bit Name	Function	RW
PCR0	Port P1 Output Format Control Bit	0: Push-pull output 1: Pseudo-N-channel open drain output (1)	RW
— (b6-b1)	No register bits; should be written with 0 and read as undefined value		—
PCE	Reserved	Should be written with 1	RW

Note:
1. This function is designated not to make the port P1 a full open drain but to turn off the P-channel transistor in the CMOS output buffer. Therefore, the absolute maximum rating of the input voltage is from -0.3 V to VCC + 0.3 V.

Figure 25.28 PCR Register

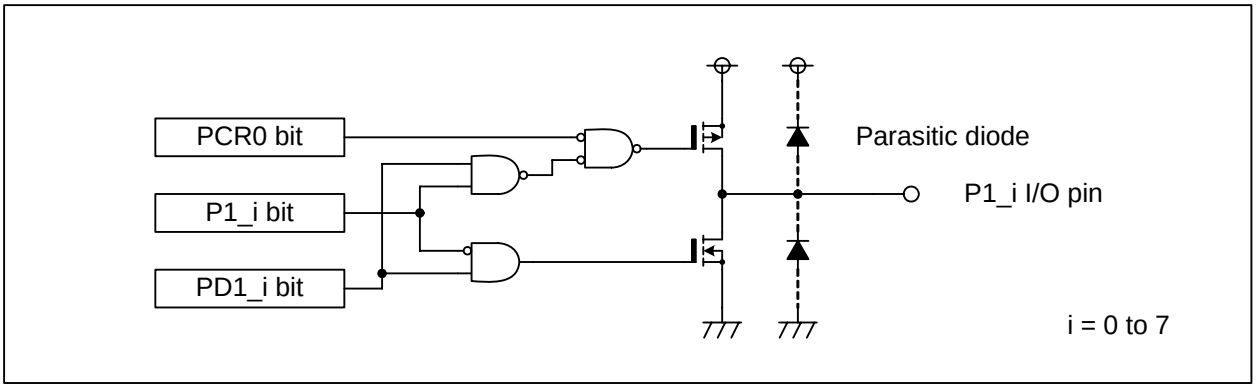


Figure 25.29 Port P1 Output Buffer Configuration

25.6 How To Configure Unused Pins

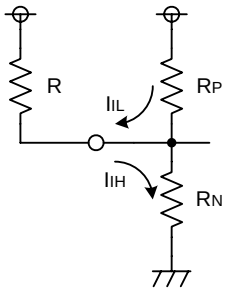
Table 25.2 and Figure 25.31 show configuring examples of unused pins on the board.

Table 25.2 Unused Pin Configuration in Single-chip Mode (1)

Pin Name	Setting
Ports P0 to P15 (excluding ports P8_5 and P14_1) (2)	Configure as input ports so that each pin is connected to VSS via its own resistor (3); or configure as output ports to leave the pins open
P14_1	Connect the pin to VSS via a resistor (3)
XOUT (4)	Leave pin open
$\overline{\text{NMI}}$ (P8_5)	Connect the pin to VCC via a resistor (3)
AVCC	Connect the pin to VCC
AVSS, VREF	Connect the pin to VSS
NSD	Connect the pin to VCC via a resistor of 1 to 4.7 kΩ

Notes:
1. Unused pins should be wired as closely as possible to the MCU (within 2 cm).
2. When configuring the pins as output ports to leave them open, note that the ports as inputs remain unchanged from when the reset is released until the mode transition is completed. During this transition, the power current may increase due to an undefined voltage level of the pins. In addition, the contents of the direction register may change because of noise or program runaway caused by the noise. To avoid these situations, reconfigure the direction register regularly by software, which may achieve the higher program reliability.
3. The resistance value appropriate to the system should be designated. The range from 10 to 100 kΩ is recommended.
4. The setting is applicable when an external clock is applied to the XIN pin

Pull-up/pull-down resistors



The figure shows the equivalent circuit of an input pin.
The equivalent input resistors (R_P and R_N) are calculated using input power current (I_{IL} and I_{IH}).

(Example) When VCC = 5.0 V, $I_{IH} = I_{IL}$ = 5 µA,

$$R_P = R_N = \frac{5.0}{5 \times 10^{-6}} = 1\ M\Omega$$

Since the voltage (V_{IH}) defined as high is more than 0.8 VCC,
the resistance value R should satisfy the following expression:

$$R // R_P : R_N = 0.2 : 0.8$$

That is,

$$R = \frac{2R_PR_N}{8R_P - 2R_N}$$

Specifically,

(Example) When VCC = 5.0 V, $I_{IH} = I_{IL}$ = 5 µA,

$$R = \frac{2 \times 10^6 \times 10^6}{8 \times 10^6 - 2 \times 10^6} = 333333$$

The maximum pull-up resistor R is approx. 330 kΩ.
The actual resistance value is the calculated value with some margins.

Figure 25.30 Pull-up/Pull-down Resistors

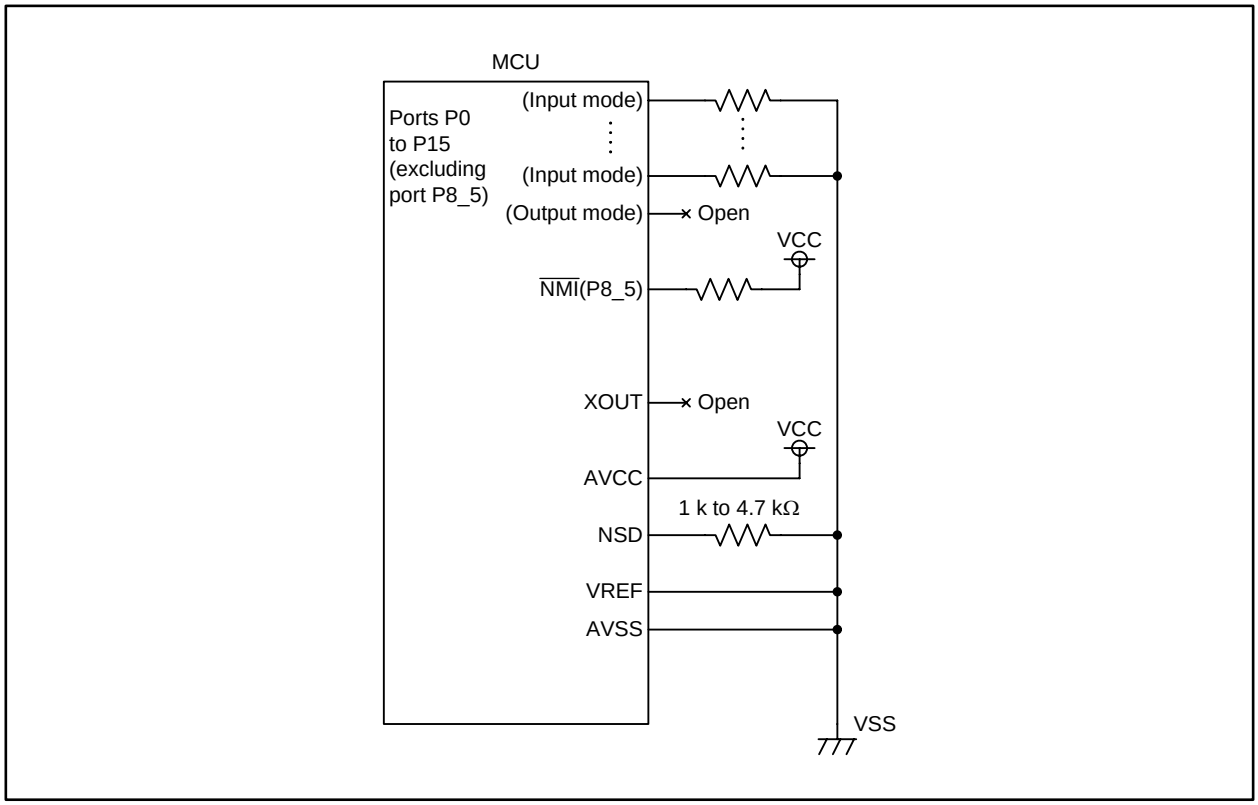


Figure 25.31 Unused Pin Configuration

26. Flash Memory

26.1 Overview

Rewrite operation to the flash memory can be performed in the following three modes: CPU rewrite mode, standard serial I/O mode, and parallel I/O mode.
Table 26.1 lists specifications of the flash memory and Table 26.2 shows the overview of each rewrite mode.

Table 26.1 Flash Memory Specifications

Item	Specification
Rewrite modes	CPU rewrite mode, standard serial I/O mode, parallel I/O mode
Structure	Block architecture. Refer to Figure 26.1
Program operation	8-byte basis
Erase operation	1-block basis
Program/erase controlled by	Software commands
Protection types	Lock bit protect, ROM code protect, ID code protect
Software commands	9 commands

Table 26.2 Flash Memory Rewrite Mode Overview

Rewrite Mode	CPU Rewrite Mode	Standard Serial I/O Mode	Parallel I/O Mode
Function	CPU executes a software command to rewrite the flash memory EW0 mode: Rewritable in areas other than the on-chip flash memory EW1 mode: Rewritable in areas other than specified blocks to be rewritten	A dedicated serial programmer rewrites the flash memory Standard serial I/O mode 1: Synchronous serial I/O selected Standard serial I/O mode 2: UART selected	A dedicated parallel programmer rewrites the flash memory
CPU operating mode	Single-chip mode	Standard serial I/O mode	Parallel I/O mode
Programmer	—	Serial programmer	Parallel programmer
On-board rewriting	Supported	Supported	Not supported

Figure 26.1 shows the on-chip flash memory structure.
The on-chip flash memory contains program area to store user programs, and data area/data flash to store the result of user programs. The program area consists of blocks 0 to 17, and data area/data flash consists of blocks A and B.
Each block can be individually protected (locked) from programing or erasing by setting the lock bit.

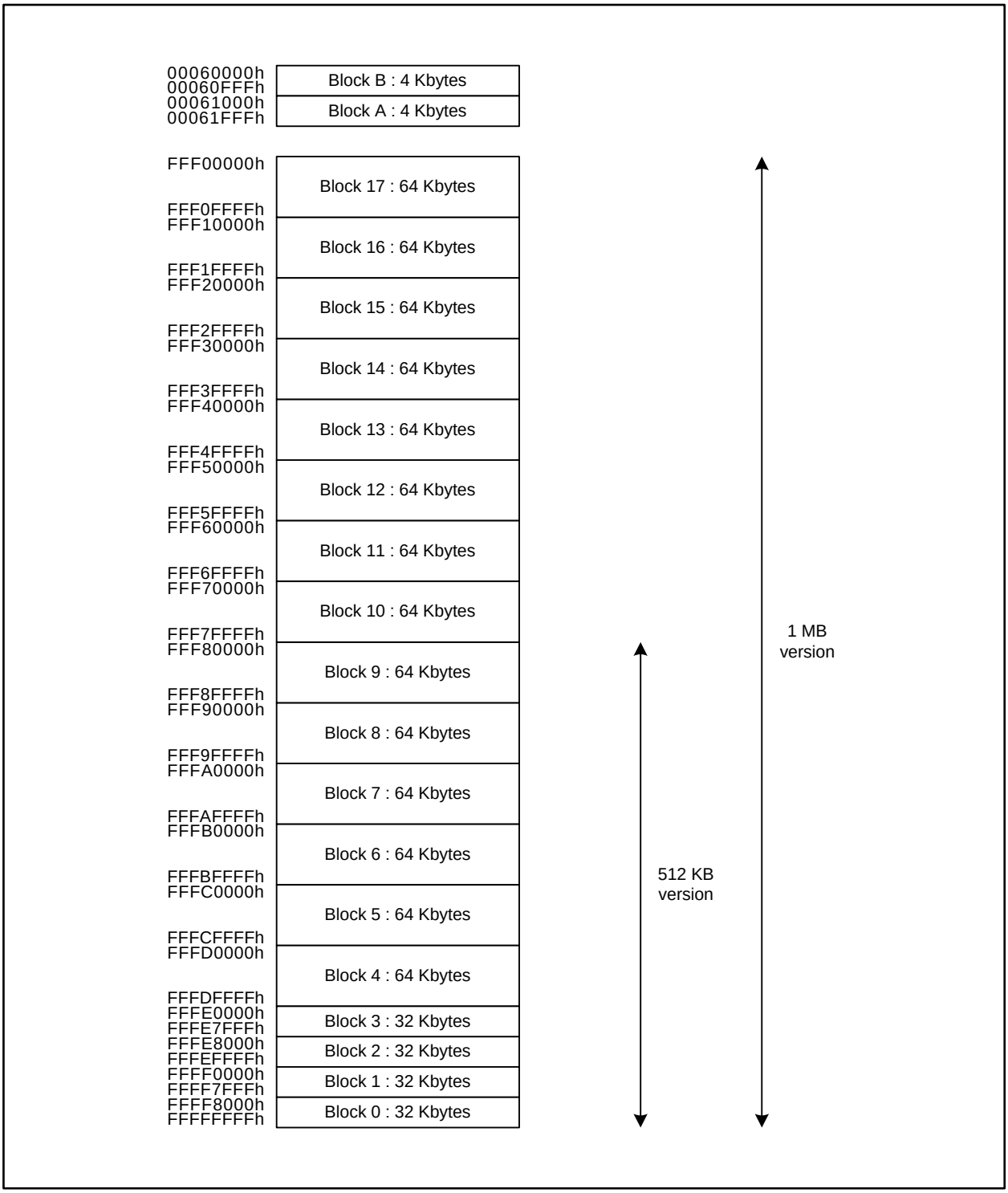


Figure 26.1 Embedded Flash Memory Block Diagram

26.2 Flash Memory Protection

There are three types of protections as shown in Table 26.3. Lock bit protection is intended to prevent accidental program or erase by program runaway. ROM code protection and ID code protection are intended to prevent read or write by a third party.

Table 26.3 Protection Types and Characteristics

Protection Type	Lock Bit Protection	ROM Code Protection	ID Code Protection
Operations to be protected	Erase, write	Read, write	Read, erase, write
Protection available in	CPU rewrite mode Standard serial I/O mode Parallel I/O mode	Parallel I/O mode	Standard serial I/O mode
Protection available for	Individual blocks	The whole flash memory	The whole flash memory
Protection activated by	Setting 0 to the lock bit of block to be protected	Setting 0 to any protect bit of blocks	Writing the program which has set an ID code to specified address
Protection deactivated by	Setting the LBD bit in the FMR register to 1 (lock bit protection disabled). Or, by erasing the blocks whose lock bits are set to 0 to permanently deactivate the protection	Erasing all blocks whose protect bits are set to 0	Inputting a proper ID code to the serial programmer

26.2.1 Lock Bit Protection

This protection is available in all three rewrite modes. When the lock bit protection is activated, all the blocks whose lock bits are set to 0 (locked) are protected against programming and erasing.

To set the lock bit to 0, the lock bit program command must be issued.

To temporarily deactivate the protection of all protected blocks, disable the lock bit protection itself by setting the LBD bit in the FMR1 register to 1 (lock bit protection disabled). To permanently deactivate the protection of a protected block, erase the respective block to set the lock bit to 1 (unlocked).

26.2.2 ROM Code Protection

This protection is available only in parallel I/O mode. When the ROM code protection is activated, the whole flash memory is protected against reading and writing.

To deactivate the protection, erase all the blocks whose protect bits are set to 0 (protected).

Each block has two protect bits. Setting any protect bit to 0 by a software command activates the protection for the whole flash memory. Table 26.4 lists protect bit addresses.

Table 26.4 Protect Bit Addresses

Block	Protect Bit 0	Protect Bit 1
Block B	00060100h	00060300h
Block A	00061100h	00061300h
Block 17	FFF00100h	FFF00300h
Block 16	FFF10100h	FFF10300h
Block 15	FFF20100h	FFF20300h
Block 14	FFF30100h	FFF30300h
Block 13	FFF40100h	FFF40300h
Block 12	FFF50100h	FFF50300h
Block 11	FFF60100h	FFF60300h
Block 10	FFF70100h	FFF70300h
Block 9	FFF80100h	FFF80300h
Block 8	FFF90100h	FFF90300h
Block 7	FFFA0100h	FFFA0300h
Block 6	FFFB0100h	FFFB0300h
Block 5	FFFC0100h	FFFC0300h
Block 4	FFFD0100h	FFFD0300h
Block 3	FFFE0100h	FFFE0300h
Block 2	FFFE8100h	FFFE8300h
Block 1	FFFF0100h	FFFF0300h
Block 0	FFFF8100h	FFFF8300h

26.2.3 ID Code Protection

This protection is available only in standard serial I/O mode. When the ID code protection is activated, a command sent from the serial programmer is accepted only if the 7-byte ID code sent from the serial programmer is identical to the ID code programmed in the flash memory. However, if the reset vector is FFFFFFFFh, the ID code check is skipped because the flash memory is considered as “erase completed”. When the reset vector is FFFFFFFFh and the ROM code protection is activated, only the block erase command is accepted.

The ID codes sent from the serial programmer are consecutively numbered as ID1, ID2, ..., and ID7. On the other hand, the ID codes programmed in the flash memory, also numbered as ID1, ID2, ..., and ID7, are respectively assigned for addresses FFFFFFE8h, FFFFFFE9h, ..., and FFFFFFEEh as shown in Figure 26.2. The ID code protection is activated when a program which has an ID code set in the corresponding address is written to the flash memory.

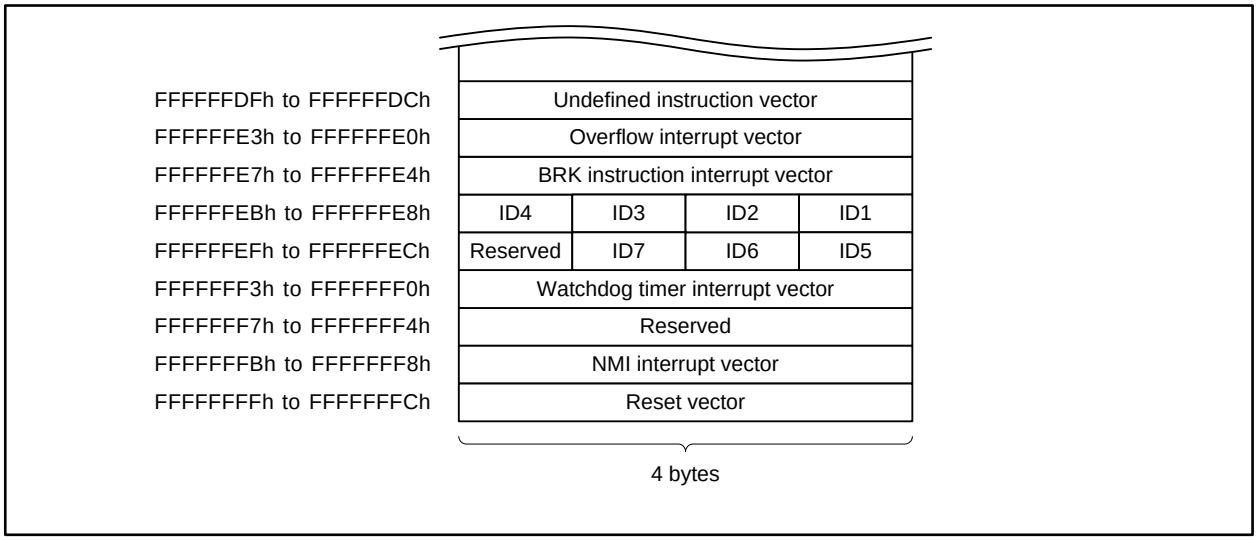


Figure 26.2 Addresses for ID Code Stored

26.3 CPU Rewrite Mode

In CPU rewrite mode, CPU executes software commands to rewrite the flash memory. The CPU accesses the flash memory not via the CPU buses but via the dedicated flash memory rewrite buses (refer to Figure 26.3).

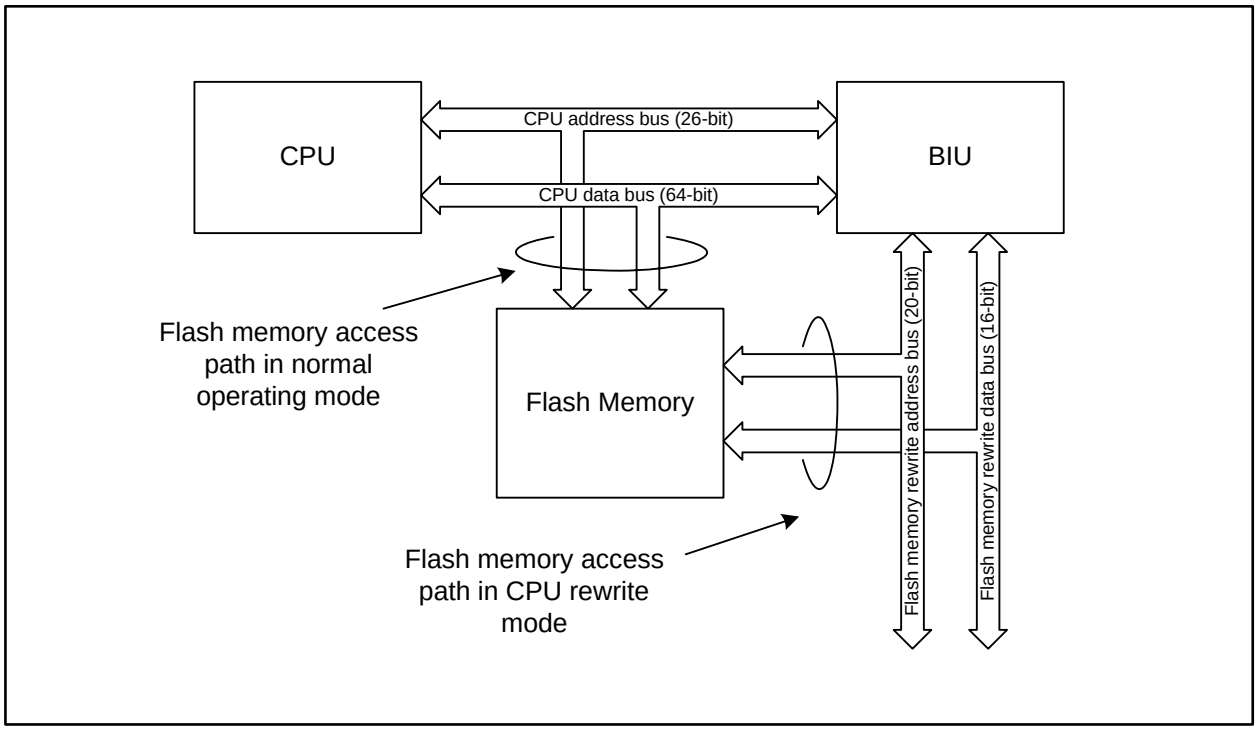

CPU
CPU address bus (26-bit)
CPU data bus (64-bit)
BIU
Flash memory access path in normal operating mode
Flash Memory
Flash memory rewrite address bus (20-bit)
Flash memory rewrite data bus (16-bit)
Flash memory access path in CPU rewrite mode

Figure 26.3 Flash Memory Access Path in CPU Rewrite Mode

Bus setting for flash memory rewrite should be performed by the FEBC register. Refer to 26.3.1 “Flash Memory Rewrite Bus Timing” and 28. “Electrical Characteristics” for the appropriate bus setting.

The CPU rewrite mode contains two sub modes: EW0 mode and EW1 mode as shown in Table 26.5.

Table 26.5 Modes EW0 and EW1

Item	EW0 Mode	EW1 Mode
Rewrite program executable spaces	Spaces other than the on-chip flash memory	Internal spaces other than specified blocks to be rewritten, internal RAM
Restriction on software command	None	• Do not execute either the program command or the block erase command for blocks where the rewrite control programs are written to • Do not execute the enter read status register mode command • Execute the enter read lock bit status mode command in RAM • Execute the enter read protect bit status mode command in RAM
Mode after program/ erase operation	Read status register mode	Read array mode
CPU state during program/erase operation	Operating	In a hold state (I/O ports maintain the state before the command was executed)
Flash memory state detection by	• Reading the FMSR0 register by a program • Executing the enter read status register mode command to read data	• Reading the FMSR0 register by a program
Other restrictions	None	• Disable interrupts (except NMI) and DMA transfer during program/erase operation

To select CPU rewrite mode, the FEW bit in the FMCR register should be set to 1. Then, EW0 mode/EW1 mode can be selected by setting the EWM bit in the FMR0 register.
Registers FMCR and FMR0 are protected by registers PRR and FPR0, respectively.
Figure 26.4 to Figure 26.12 show associated registers.

Flash Memory Control Register (1)

b7	b6	b5	b4	b3	b2	b1	b0
	0	0	0	0	0	0	1

Symbol: FMCR
Address: 0006h
Reset Value: 0000 0001b

Bit Symbol	Bit Name	Function	RW
— (b0)	Reserved	Should be written with 1	RW
— (b6-b1)	Reserved	Should be written with 0	RW
FEW	CPU Rewrite Mode Setting Bit (2)	0: Normal operating mode 1: CPU rewrite mode	RW

Notes:
1. Set the PRR register to AAh (write enabled) before rewriting this register.
2. When the MRS bit in the VRCR register is 1 (main regulator stopped), this bit should not be set to 1.

Figure 26.4 FMCR Register

Flash Memory Rewrite Bus Control Register (1)

b15 b8 b7 b0: 0 1 0 1 [ ] [ ] 0 [ ] [ ] [ ] [ ] [ ] [ ] [ ] [ ]

Symbol	Address	Reset Value
FEBC	001Dh-001Ch	0000h

Bit Symbol	Bit Name	Function	RW
FWR0	$\overline{RD}$ Pulse Width Setting Bit	b3 b2 b1 b0 0 0 0 0 : *wr* = 1 0 0 0 1 : *wr* = 2 0 1 0 1 : *wr* = 3 0 1 1 0 : *wr* = 4 1 0 1 0 : *wr* = 5 1 0 1 1 : *wr* = 6 1 1 1 1 : *wr* = 7 Only use the combinations listed above	RW
FWR1			RW
FWR2			RW
FWR3			RW
FWR4	$\overline{RD}$ Pulse Width Extension Select Bit	0: No pulse width extension 1: Pulse width extension selected	RW
— (b5)	Reserved	Should be written with 0	RW
MPY0	Multiplied Cycle Setting Bit	b7 b6 0 0 : Do not use this combination 0 1 : Do not use this combination 1 0 : *mpy* = 3 1 1 : *mpy* = 4	RW
MPY1			RW
FSUW0	Address Setup Before $\overline{WR}$ Setting Bit	b9 b8 0 0 : *suw* = 0 0 1 : *suw* = 1 1 0 : *suw* = 2 1 1 : *suw* = 3	RW
FSUW1			
FWW0	$\overline{WR}$ Pulse Width Setting Bit	b11 b10 0 0 : *ww* = 1 0 1 : *ww* = 2 1 0 : *ww* = 3 1 1 : *ww* = 4	RW
FWW1			
— (b12)	Reserved	Should be written with 1	RW
— (b13)	Reserved	Should be written with 0	RW
— (b14)	Reserved	Should be written with 1	RW
— (b15)	Reserved	Should be written with 0	RW

Note:
1. Set the PRR register to AAh (write enabled) before rewriting this register.

Figure 26.5 FEBC Register

Flash Register Protection Unlock Register 0

b7	b6	b5	b4	b3	b2	b1	b0
0	0	0	0	0	0	0	

Symbol	Address	Reset Value
FPR0	40008h	0000 0000b

Bit Symbol	Bit Name	Function	RW
PR0	Protection Unlock Bit	To registers FMR0 and FMR1, 0: Write disabled 1: Write enabled	RW
— (b7-b1)	Reserved	Should be written with 0	RW

Figure 26.6 FPR0 Register

Flash Memory Control Register 0 (1, 2)

b7	b6	b5	b4	b3	b2	b1	b0
0		0					

Symbol	Address	Reset Value
FMR0	40000h	0X01 XX00b

Bit Symbol	Bit Name	Function	RW
EWM	Rewrite Mode Select Bit	0: EW0 mode 1: EW1 mode	RW
LBM	Lock Bit Read Mode Setting Bit	0: Read via data bus (3) 1: Read by the LBS bit (4)	RW
LBS	Lock Bit Status Flag (4)	0: Locked 1: Unlocked	RO
RRDY	Read Ready Flag	0: Busy 1: Ready	RO
FCA	Final Command Accept Busy Flag	0: Final command accept ready 1: Final command accept busy	RO
— (b5)	Reserved	Should be written with 0	RW
— (b6)	Reserved	This bit is read as undefined value	RO
— (b7)	Reserved	Should be written with 0	RW

Notes:
1. Set the PR0 bit in the FPR0 register to 1 (write enabled) before rewriting this register.
2. This register is reset after exiting wait mode or stop mode.
3. After entering read lock bit status mode, if any even address in the corresponding block is read, the lock bit status is indicated in bit 6 of read data.
4. The LBS bit indicates the lock bit status by the read lock bit status command.

Figure 26.7 FMR0 Register

Flash Memory Control Register 1 (1)

b7	b6	b5	b4	b3	b2	b1	b0
0	0	0	0		0		0

Symbol: FMR1 — Address: 40009h — Reset Value: 0000 0010b

Bit Symbol	Bit Name	Function	RW
— (b0)	Reserved	Should be written with 0	RW
RR	Reset Release Bit	0: Reset 1: Reset released	RW
— (b2)	Reserved	Should be written with 0	RW
LBD	Lock Bit Protect Disable Bit	0: Lock bit protection enabled 1: Lock bit protection disabled	RW
— (b7-b4)	Reserved	Should be written with 0	RW

Note:
1. Set the PR0 bit in the FPR0 register to 1 (write enabled) before rewriting this register.

Figure 26.8 FMR1 Register

Flash Memory Status Register 0

b7	b6	b5	b4	b3	b2	b1	b0

Symbol: FMSR0 — Address: 40001h — Reset Value: 1000 0000b

Bit Symbol	Bit Name	Function	RW
— (b3-b0)	Reserved	These bits are read as undefined value	RO
WERR	Program Error Flag	0: No program error 1: Program error occurred	RO
EERR	Erase Error Flag	0: No erase error 1: Erase error occurred (1)	RO
— (b6)	Reserved	This bit is read as undefined value	RO
RDY	Ready Flag	0: Busy 1: Ready	RO

Note:
1. If an erase error has occurred, issue the clear status register command first, then re-issue the block erase command repeatedly until no more erase error occurs. After that, execute three more extra block erase operations.

Figure 26.9 FMSR0 Register

Block Protect Bit Monitor Register 0 (1)

b7 b6 b5 b4 b3 b2 b1 b0

Symbol	Address	Reset Value
FBPM0	4000Ah	??X? ????b (1)

Bit Symbol	Bit Name	Function	RW
BP0	Block0 Protect Bit Monitor Flag	0: Protected 1: Protection unlocked	RO
BP1	Block1 Protect Bit Monitor Flag	0: Protected 1: Protection unlocked	RO
BP2	Block2 Protect Bit Monitor Flag	0: Protected 1: Protection unlocked	RO
BP3	Block3 Protect Bit Monitor Flag	0: Protected 1: Protection unlocked	RO
BP4	Block4 Protect Bit Monitor Flag	0: Protected 1: Protection unlocked	RO
— (b5)	Reserved	This bit is read as undefined value	RO
BP5	Block5 Protect Bit Monitor Flag	0: Protected 1: Protection unlocked	RO
BP6	Block6 Protect Bit Monitor Flag	0: Protected 1: Protection unlocked	RO

Note:
1. This register is updated only once after exiting reset mode. The protect bit status at that time is applied as reset value.

Figure 26.10 FBPM0 Register

Block Protect Bit Monitor Register 1 (1)

b7 b6 b5 b4 b3 b2 b1 b0

Symbol	Address	Reset Value
FBPM1	4000Bh	XXX? ????b (1)

Bit Symbol	Bit Name	Function	RW
BP7	Block7 Protect Bit Monitor Flag	0: Protected 1: Protection unlocked	RO
BP8	Block8 Protect Bit Monitor Flag	0: Protected 1: Protection unlocked	RO
BP9	Block9 Protect Bit Monitor Flag	0: Protected 1: Protection unlocked	RO
BPB	BlockB Protect Bit Monitor Flag	0: Protected 1: Protection unlocked	RO
BPA	BlockA Protect Bit Monitor Flag	0: Protected 1: Protection unlocked	RO
— (b7-b5)	Reserved	These bits are read as undefined value	RO

Note:
1. This register is updated only once after exiting reset mode. The protect bit status at that time is applied as reset value.

Figure 26.11 FBPM1 Register

Block Protect Bit Monitor Register 2 (1)

b7 b6 b5 b4 b3 b2 b1 b0

Symbol	Address	Reset Value
FBPM2	40011h	???? ????b (1)

Bit Symbol	Bit Name	Function	RW
BP10	Block10 Protect Bit Monitor Flag	0: Protected 1: Protection unlocked	RO
BP11	Block11 Protect Bit Monitor Flag	0: Protected 1: Protection unlocked	RO
BP12	Block12 Protect Bit Monitor Flag	0: Protected 1: Protection unlocked	RO
BP13	Block13 Protect Bit Monitor Flag	0: Protected 1: Protection unlocked	RO
BP14	Block14 Protect Bit Monitor Flag	0: Protected 1: Protection unlocked	RO
BP15	Block15 Protect Bit Monitor Flag	0: Protected 1: Protection unlocked	RO
BP16	Block16 Protect Bit Monitor Flag	0: Protected 1: Protection unlocked	RO
BP17	Block17 Protect Bit Monitor Flag	0: Protected 1: Protection unlocked	RO

Note:
1. This register is updated only once after exiting reset mode. The protect bit status at that time is applied as reset value.

Figure 26.12 FBPM2 Register

26.3.1 Flash Memory Rewrite Bus Timing

The bus setting for the flash memory rewrite is performed by using the FEBC register. This section specifically describes the setting of FEBC register.

The reference clock is the base clock set using bits BCD1 and BCD0 in the CCR register. Time duration including tsu, tw, tc and th are specified by base clock cycles.

Table 26.6 to Table 26.8 show the correlation of read cycle and setting of following bits: MPY1, MPY0, and FWR4 to FWR0, according to respective peripheral bus clock divide ratio. Table 26.9 to Table 26.11 show the correlation of write cycle and setting of following bits: MPY1, MPY0, FSUW1, FSUW0, FWW1, and FWW0. Associated read/write timings are respectively illustrated in Figure 26.13 and Figure 26.14.

Read/write cycle timing is selected from these tables below to meet the timing requirements in CPU rewrite mode described in the electrical characteristics.

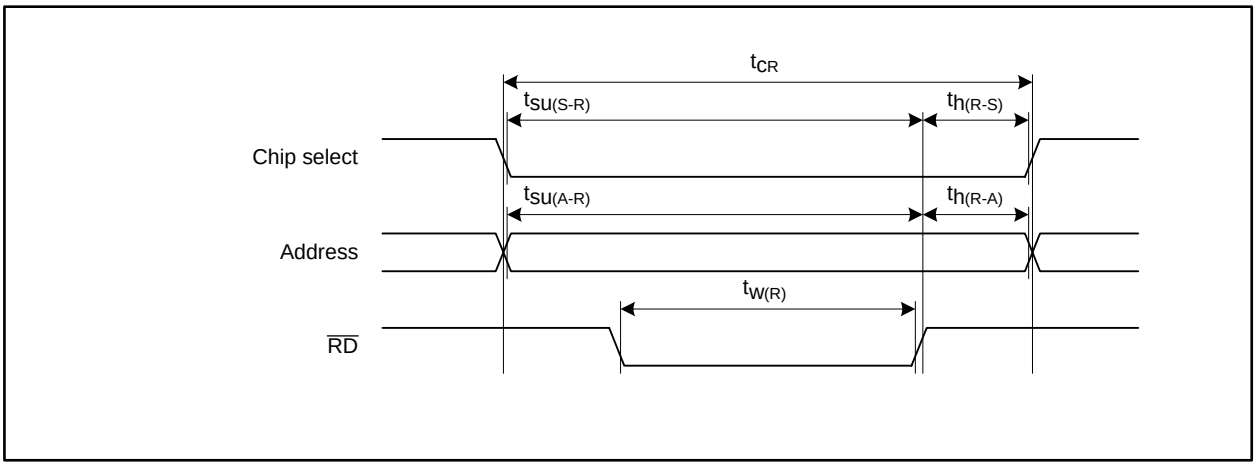


Figure 26.13 Read Timing

Table 26.6 The Read Cycle and Bit Settings: MPY1, MPY0, and FWR4 to FWR0, When Peripheral Bus Clock is Divided by 2 (unit: cycles)

FWR3 to FWR0 Bit Settings		FWR4 Bit Settings	MPY1 and MPY0 bit settings							
			10b				11b			
			mpy = 3				*mpy* = 4			
			tsu(S-R), tsu(A-R)	tw(R)	tcR	th(R-S), th(R-A)	tsu(S-R), tsu(A-R)	tw(R)	tcR	th(R-S), th(R-A)
0000b	*wr* = 1	0	4	3	4	0	6	5	6	0
		1	6	5	6	0	6	5	6	0
0001b	*wr* = 2	0	8	7	8	0	10	9	10	0
		1	8	7	8	0	10	9	10	0
0101b	*wr* = 3	0	10	9	10	0	14	13	14	0
		1	12	11	12	0	14	13	14	0
0110b	*wr* = 4	0	14	13	14	0	18	17	18	0
		1	14	13	14	0	18	17	18	0
1010b	*wr* = 5	0	16	15	16	0	22	21	22	0
		1	18	17	18	0	22	21	22	0
1011b	*wr* = 6	0	20	19	20	0	26	25	26	0
		1	20	19	20	0	26	25	26	0
1111b	*wr* = 7	0	22	21	22	0	30	29	30	0
		1	24	23	24	0	30	29	30	0

Table 26.7 The Read Cycle and Bit Settings: MPY1, MPY0, and FWR4 to FWR0, When Peripheral Bus Clock is Divided by 3 (unit: cycles)

FWR3 to FWR0 Bit Settings		FWR4 Bit Settings	MPY1 and MPY0 bit settings							
			10b				11b			
			mpy = 3				*mpy* = 4			
			tsu(S-R), tsu(A-R)	tw(R)	tcR	th(R-S), th(R-A)	tsu(S-R), tsu(A-R)	tw(R)	tcR	th(R-S), th(R-A)
0000b	*wr* = 1	0	6	4.5	6	0	6	4.5	6	0
		1	6	4.5	6	0	6	4.5	6	0
0001b	*wr* = 2	0	9	7.5	9	0	9	7.5	9	0
		1	9	7.5	9	0	12	10.5	12	0
0101b	*wr* = 3	0	12	10.5	12	0	15	13.5	15	0
		1	12	10.5	12	0	15	13.5	15	0
0110b	*wr* = 4	0	15	13.5	15	0	18	16.5	18	0
		1	15	13.5	15	0	18	16.5	18	0
1010b	*wr* = 5	0	18	16.5	18	0	21	19.5	21	0
		1	18	16.5	18	0	24	22.5	24	0
1011b	*wr* = 6	0	21	19.5	21	0	27	25.5	27	0
		1	21	19.5	21	0	27	25.5	27	0
1111b	*wr* = 7	0	24	22.5	24	0	30	28.5	30	0
		1	24	22.5	24	0	30	28.5	30	0

Table 26.8 The Read Cycle and Bit Settings: MPY1, MPY0, and FWR4 to FWR0, When Peripheral Bus Clock is Divided by 4 (unit: cycles)

FWR3 to FWR0 Bit Settings		FWR4 Bit Settings	MPY1 and MPY0 bit settings							
			10b				11b			
			mpy = 3				*mpy* = 4			
			tsu(S-R), tsu(A-R)	tw(R)	tcR	th(R-S), th(R-A)	tsu(S-R), tsu(A-R)	tw(R)	tcR	th(R-S), th(R-A)
0000b	*wr* = 1	0	4	2	4	0	8	6	8	0
		1	8	6	8	0	8	6	8	0
0001b	*wr* = 2	0	8	6	8	0	12	10	12	0
		1	8	6	8	0	12	10	12	0
0101b	*wr* = 3	0	12	10	12	0	16	14	16	0
		1	12	10	12	0	16	14	16	0
0110b	*wr* = 4	0	16	14	16	0	20	18	20	0
		1	16	14	16	0	20	18	20	0
1010b	*wr* = 5	0	16	14	16	0	24	22	24	0
		1	20	18	20	0	24	22	24	0
1011b	*wr* = 6	0	20	18	20	0	28	26	28	0
		1	20	18	20	0	28	26	28	0
1111b	*wr* = 7	0	24	22	24	0	32	30	32	0
		1	24	22	24	0	32	30	32	0

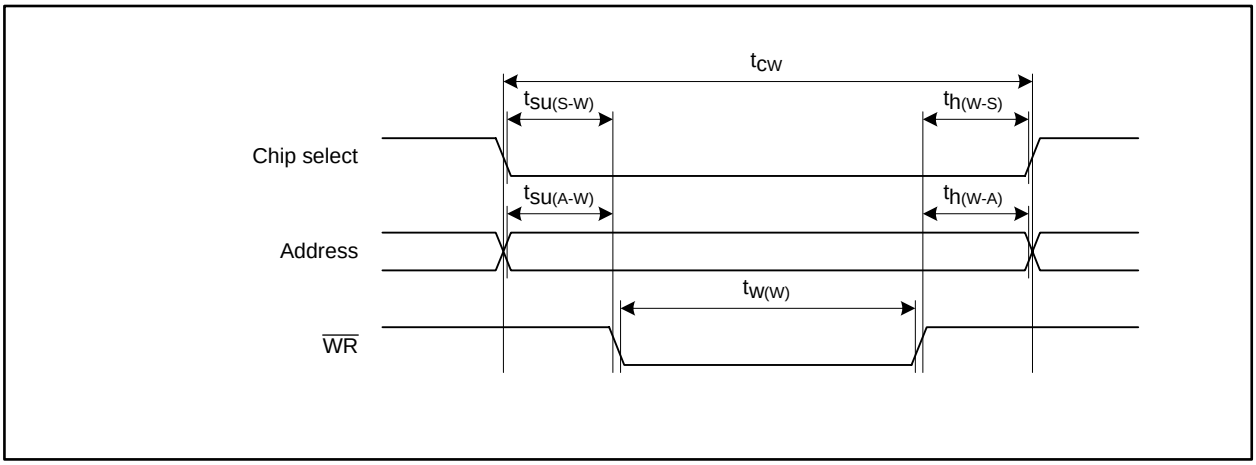


Figure 26.14 Write Timing

Table 26.9 The Write Cycle and Bit Settings: MPY1, MPY0, FSUW1, FSUW0, FWW1, and FWW0, When Peripheral Bus Clock is Divided by 2 (unit: cycles)

FSUW1 and FSUW0 Bit Settings		FWW1and FWW0 Bit Settings		MPY1 and MPY0 Bit Settings							
				10b				11b			
				mpy = 3				*mpy* = 4			
				tsu(S-W), tsu(A-W)	tw(W)	tcw	th(W-S), th(W-A)	tsu(S-W), tsu(A-W)	tw(W)	tcw	th(W-S), th(W-A)
00b	*suw* = 0	00b	*ww* = 1	1	3	6	2	1	4	6	1
		01b	*ww* = 2	1	6	8	1	1	8	10	1
		10b	*ww* = 3	1	9	12	2	1	12	14	1
		11b	*ww* = 4	1	12	14	1	1	16	18	1
01b	*suw* = 1	00b	*ww* = 1	4	3	8	1	5	4	10	1
		01b	*ww* = 2	4	6	12	2	5	8	14	1
		10b	*ww* = 3	4	9	14	1	5	12	18	1
		11b	*ww* = 4	4	12	18	2	5	16	22	1
10b	*suw* = 2	00b	*ww* = 1	7	3	12	2	9	4	14	1
		01b	*ww* = 2	7	6	14	1	9	8	18	1
		10b	*ww* = 3	7	9	18	2	9	12	22	1
		11b	*ww* = 4	7	12	20	1	9	16	26	1
11b	*suw* = 3	00b	*ww* = 1	10	3	14	1	13	4	18	1
		01b	*ww* = 2	10	6	18	2	13	8	22	1
		10b	*ww* = 3	10	9	20	1	13	12	26	1
		11b	*ww* = 4	10	12	24	2	13	16	30	1

Table 26.10 The Write Cycle and Bit Settings: MPY1, MPY0, FSUW1, FSUW0, FWW1, and FWW0, When Peripheral Bus Clock is Divided by 3 (unit: cycles)

FSUW1 and FSUW0 Bit Settings		FWW1and FWW0 Bit Settings		MPY1 and MPY0 Bit Settings							
				10b				11b			
				mpy = 3				*mpy* = 4			
				tsu(S-W), tsu(A-W)	tw(W)	tcw	th(W-S), th(W-A)	tsu(S-W), tsu(A-W)	tw(W)	tcw	th(W-S), th(W-A)
00b	*suw* = 0	00b	*ww* = 1	1	3	6	2	1	4	6	1
		01b	*ww* = 2	1	6	9	2	1	8	12	3
		10b	*ww* = 3	1	9	12	2	1	12	15	2
		11b	*ww* = 4	1	12	15	2	1	16	18	1
01b	*suw* = 1	00b	*ww* = 1	4	3	9	2	6	3	12	3
		01b	*ww* = 2	4	6	12	2	6	7	15	2
		10b	*ww* = 3	4	9	15	2	6	11	18	1
		11b	*ww* = 4	4	12	18	2	6	15	24	3
10b	*suw* = 2	00b	*ww* = 1	7	3	12	2	9	4	15	2
		01b	*ww* = 2	7	6	15	2	9	8	18	1
		10b	*ww* = 3	7	9	18	2	9	12	24	3
		11b	*ww* = 4	7	12	21	2	9	16	27	2
11b	*suw* = 3	00b	*ww* = 1	10	3	15	2	13	4	18	1
		01b	*ww* = 2	10	6	18	2	13	8	24	3
		10b	*ww* = 3	10	9	21	2	13	12	27	2
		11b	*ww* = 4	10	12	24	2	13	16	30	1

Table 26.11 The Write Cycle and Bit Settings: MPY1, MPY0, FSUW1, FSUW0, FWW1, and FWW0, When Peripheral Bus Clock is Divided by 4 (unit: cycles)

FSUW1 and FSUW0 Bit Settings		FWW1and FWW0 Bit Settings		MPY1 and MPY0 Bit Settings							
				10b				11b			
				mpy = 3				*mpy* = 4			
				tsu(S-W), tsu(A-W)	tw(W)	tcw	th(W-S), th(W-A)	tsu(S-W), tsu(A-W)	tw(W)	tcw	th(W-S), th(W-A)
00b	*suw* = 0	00b	*ww* = 1	1	3	8	4	1	4	8	3
		01b	*ww* = 2	1	6	8	1	1	8	12	3
		10b	*ww* = 3	1	9	12	2	1	12	16	3
		11b	*ww* = 4	1	12	16	3	1	16	20	3
01b	*suw* = 1	00b	*ww* = 1	4	3	8	1	5	4	12	3
		01b	*ww* = 2	4	6	12	2	5	8	16	3
		10b	*ww* = 3	4	9	16	3	5	12	20	3
		11b	*ww* = 4	4	12	20	4	5	16	24	3
10b	*suw* = 2	00b	*ww* = 1	8	2	12	2	9	4	16	3
		01b	*ww* = 2	8	5	16	3	9	8	20	3
		10b	*ww* = 3	8	8	20	4	9	12	24	3
		11b	*ww* = 4	8	11	20	1	9	16	28	3
11b	*suw* = 3	00b	*ww* = 1	10	3	16	3	13	4	20	3
		01b	*ww* = 2	10	6	20	4	13	8	24	3
		10b	*ww* = 3	10	9	20	1	13	12	28	3
		11b	*ww* = 4	10	12	24	2	13	16	32	3

26.3.2 Software Commands

In CPU rewrite mode, software commands enable to rewrite or erase the flash memory. A write of command and read of data should be performed in 16-bit units.
Table 26.12 lists the software commands.

Table 26.12 Software Commands

Command	First Command Cycle		Second Command Cycle	
	Address	Data	Address	Data
Enter read array mode	FFFFF800h	00FFh	—	—
Enter read status register mode (1)	FFFFF800h	0070h	—	—
Clear status register	FFFFF800h	0050h	—	—
Program (2)	FFFFF800h	0043h	WA	WD
Block erase	FFFFF800h	0020h	BA	00D0h
Lock bit program	FFFFF800h	0077h	BA	00D0h
Read lock bit status	FFFFF800h	0071h	BA	00D0h
Enter read lock bit status mode (3)	FFFFF800h	0071h	—	—
Protect bit program	FFFFF800h	0067h	PBA	00D0h
Enter read protect bit status mode (3)	FFFFF800h	0061h	—	—

WA: Even address to be written
WD: 16-bit data to be written
BA: An even address within a specific block
PBA: Protect bit address (Refer to Table 26.4)

Notes:
1. This command cannot be executed in EW1 mode.
2. A set of command consists of five words from the first command to the fifth. The program is performed in 64-bit (four-word) unit. The higher 29 bits of the address WA should be fixed and the lower three bits of respective commands from the second to fifth should be set to 000b, 010b, 100b, and 110b for the addresses 0h, 2h, 4h, and 6h, or 8h, Ah, Ch, and Eh.
3. This command should be executed in RAM.

26.3.3 Mode Transition

CPU rewrite mode supports four flash memory operating modes:

- Read array mode
- Read status register mode
- Read lock bit status mode
- Read protect bit status mode

When reading the flash memory in these modes, the content of memory, the content of status register, the state of lock bit of read block, and the state of protect bit are respectively read. The details are listed in Table 26.13 to Table 26.15.

Table 26.13 Status Register

Bit	Bit Symbol	Bit Name	Definition	
			0	1
b15-b8	—	Disabled bit	—	—
b7	SR7	Sequencer status	BUSY	READY
b6	—	Reserved bit	—	—
b5	SR5	Erase status	Successfully completed	Error
b4	SR4	Program status	Successfully completed	Error
b3	—	Reserved bit	—	—
b2	—	Reserved bit	—	—
b1	—	Reserved bit	—	—
b0	—	Reserved bit	—	—

Table 26.14 Lock Bit Status

Bit	Bit Symbol	Bit Name	Definition	
			0	1
b15-b7	—	Disabled bit	—	—
b6	LBS	Lock bit status	Locked	Unlocked
b5-b0	—	Disabled bit	—	—

Table 26.15 Protect Bit Status

Bit	Bit Symbol	Bit Name	Definition	
			0	1
b15-b7	—	Disabled bit	—	—
b6	PBS	Protect bit status	Protected	Unprotected
b5-b0	—	Disabled bit	—	—

In these operating modes, a program or erase operation can be performed by software commands. After the operation is completed, the flash memory module automatically enters read array mode (in EW1 mode) or read status register mode (in EW0 mode).

26.3.4 How to Issue Software Commands

This section describes how to issue the software commands.
These commands should be issued while the RDY bit in the FMSR0 register is 1 (ready).

26.3.4.1 Enter Read Array Mode Command

This command is executed to enter read array mode.
When 00FFh is written to address FFFFF800h, the flash memory enters read array mode. In this mode, data stored to a given address in memory can be read.
In EW1 mode, the flash memory is always in read array mode.

26.3.4.2 Enter Read Status Register Mode

This command is executed to enter read status register mode.
When 0070h is written to address FFFFF800h, data of the status register is read in any address of the flash memory.
Do not execute this command in EW1 mode.

26.3.4.3 Clear Status Register

This command is executed to reset the status register in the flash memory.
When 0050h is written to address FFFFF800h, bits SR5 and SR4 in the status register become 0 (successfully completed) (Refer to Table 26.13). Consequently, bits EERR and WERR in the FMSR0 register become 0 (no errors).

26.3.4.4 Program Command

This command is executed to program the flash memory in eight-byte (four-word) unit.
To start automatic programming (program and program-verify operation), write 0043h to address FFFFF800h, then write data to addresses 8n + 0 to 8n + 6. Verify that the FCA bit in the FMR0 register is 0 just before executing the final command.
To monitor the automatic program operation, read the RDY bit in the FMSR0 register. This bit indicates 0 (busy) when the operation is in progress and 1 (ready) when the operation is completed.
The operation result can be verified by the WERR bit in the FMSR0 register (Refer to 26.3.5 “Status Check”).
Do not write additional data to the address already programmed.

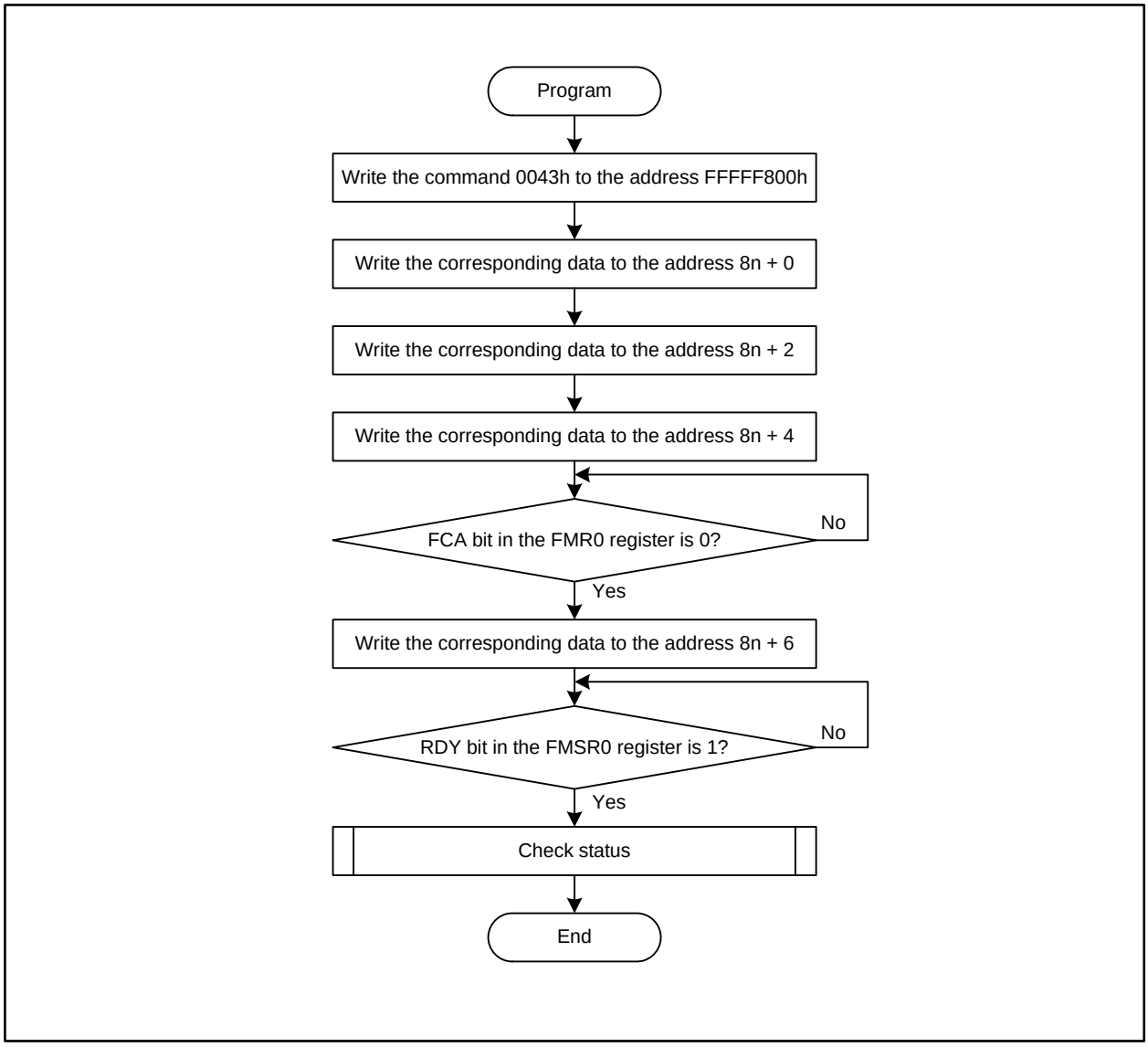

Program
Write the command 0043h to the address FFFFF800h
Write the corresponding data to the address 8n + 0
Write the corresponding data to the address 8n + 2
Write the corresponding data to the address 8n + 4
FCA bit in the FMR0 register is 0?
No
Yes
Write the corresponding data to the address 8n + 6
RDY bit in the FMSR0 register is 1?
No
Yes
Check status
End

Figure 26.15 Program Command Flow

26.3.4.5 Block Erase Command

This command is executed to erase a specified block in the flash memory.

To start automatic erasing of the specified block (erase and erase-verify operation), write 0020h to address FFFFF800h, verify that the FCA bit in the FMR0 register is 0, then write 00D0h to an even address of the corresponding block.

To monitor the automatic erase operation, read the RDY bit in the FMSR0 register. This bit indicates 0 (busy) when the operation is in progress and 1 (ready) when the operation is completed.

The operation result can be verified by the EERR bit in the FMSR0 register (Refer to 26.3.5 “Status Check”).

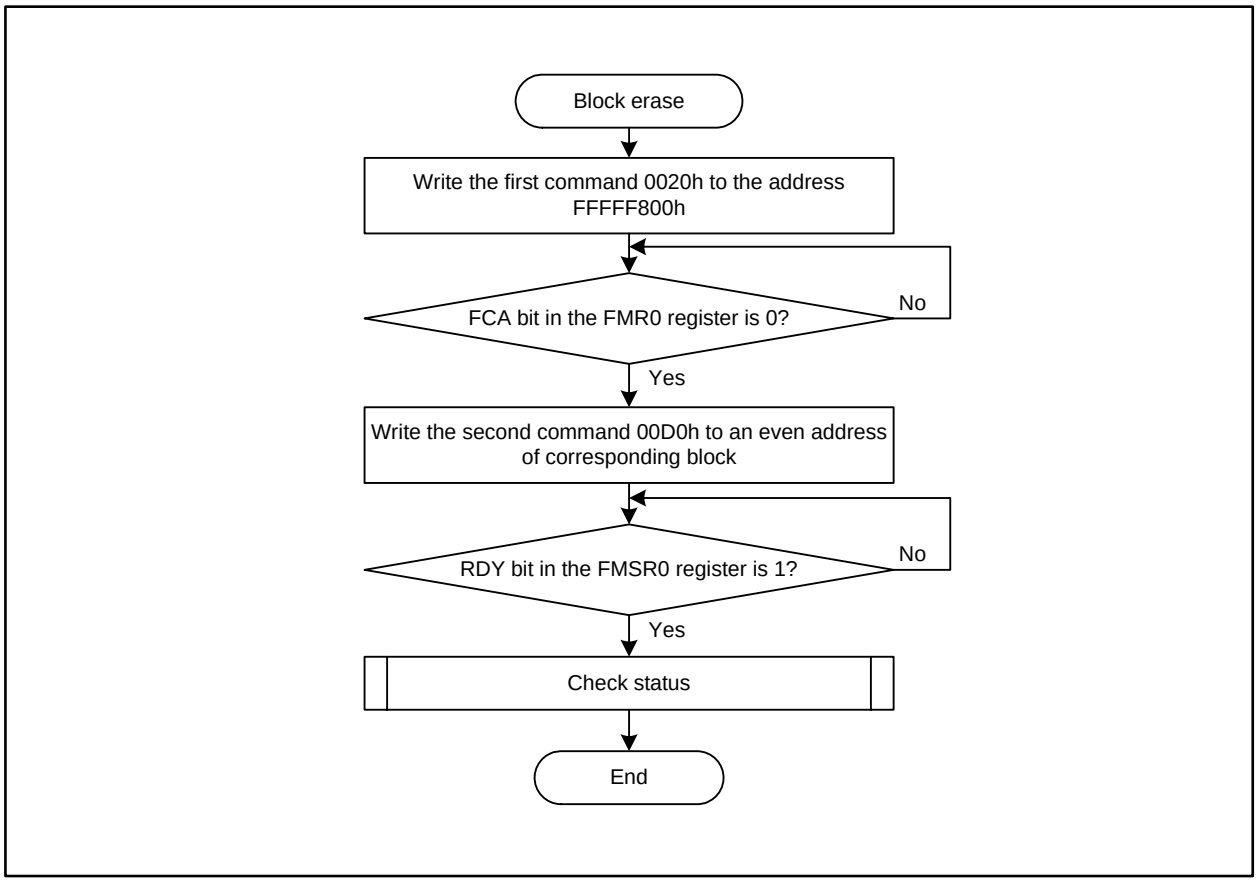

Block erase
Write the first command 0020h to the address FFFFF800h
FCA bit in the FMR0 register is 0?
No
Yes
Write the second command 00D0h to an even address of corresponding block
RDY bit in the FMSR0 register is 1?
No
Yes
Check status
End

Figure 26.16 Block Erase Command Flow

26.3.4.6 Lock Bit Program Command

This command is executed to lock a specified block in the flash memory.
To lock the block, write 0077h to address FFFFF800h, verify that the FCA bit in the FMR0 register is 0, then write 00D0h to an even address of the corresponding block. Then the lock bit of the block becomes 0 (locked).
To monitor the lock bit program, read the RDY bit in the FMSR0 register. This bit indicates 0 (busy) when the operation is in progress and 1 (ready) when the operation is completed.
The state of lock bit can be verified by the read lock bit status command if the LBM bit in the FMR0 register is 1 (read by the LBS bit) (Refer to 26.3.4.7 “Read Lock Bit Status Command”). If the LBM bit is 0 (read via data bus), enter read lock bit status mode (Refer to 26.3.4.8 “Enter Read Lock Bit Status Mode Command”).

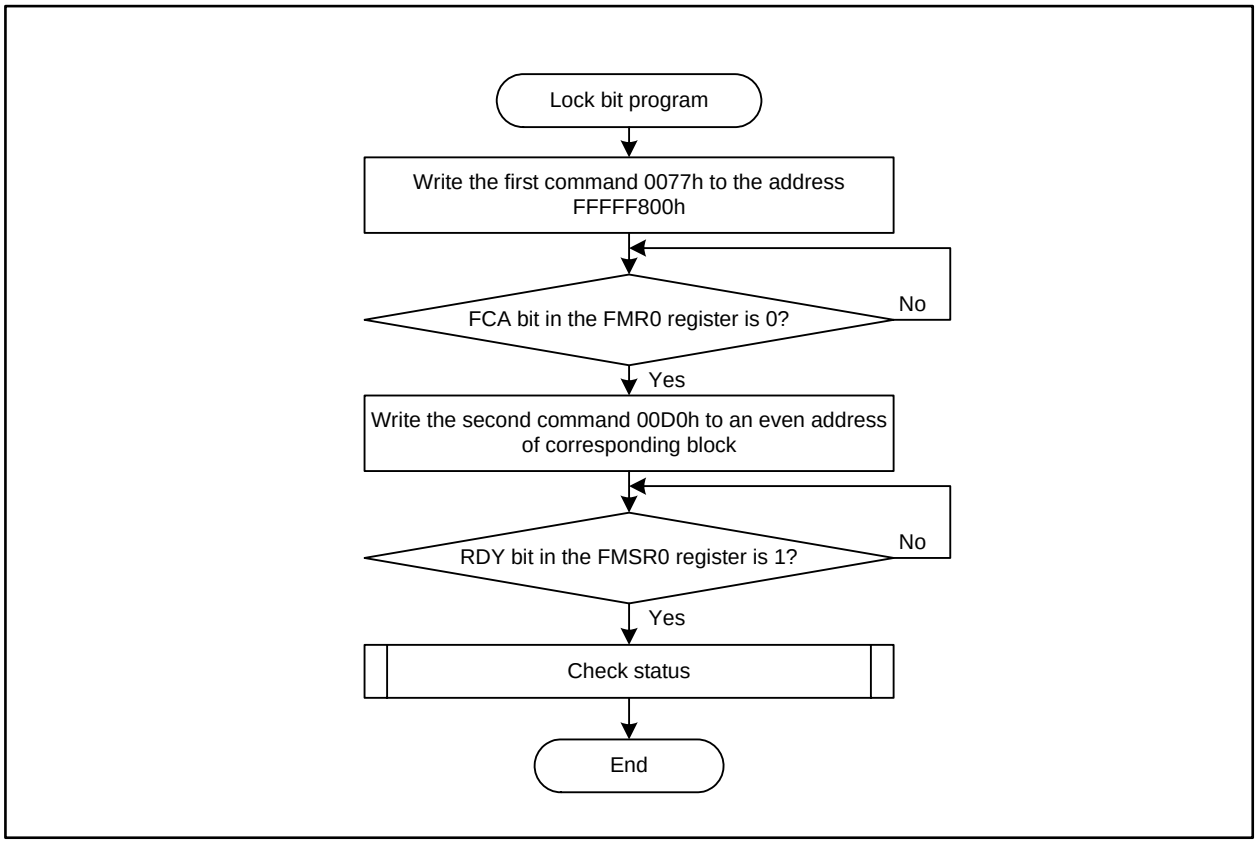

Lock bit program
Write the first command 0077h to the address FFFFF800h
FCA bit in the FMR0 register is 0?
No
Yes
Write the second command 00D0h to an even address of corresponding block
RDY bit in the FMSR0 register is 1?
No
Yes
Check status
End

Figure 26.17 Lock Bit Program Command Flow

26.3.4.7 Read Lock Bit Status Command

This command is executed to verify if a specified block in the flash memory is locked. This command is available when the LBM bit in the FMR0 register is 1 (read by the LBS bit).
To read the LBS bit from the FMR0 register, write 0071h to address FFFFF800h, verify that the FCA bit in the FMR0 register is 0, then write 00D0h to an even address of the corresponding block.
Read the LBS bit after the RDY bit in the FMSR0 register becomes 1 (ready).

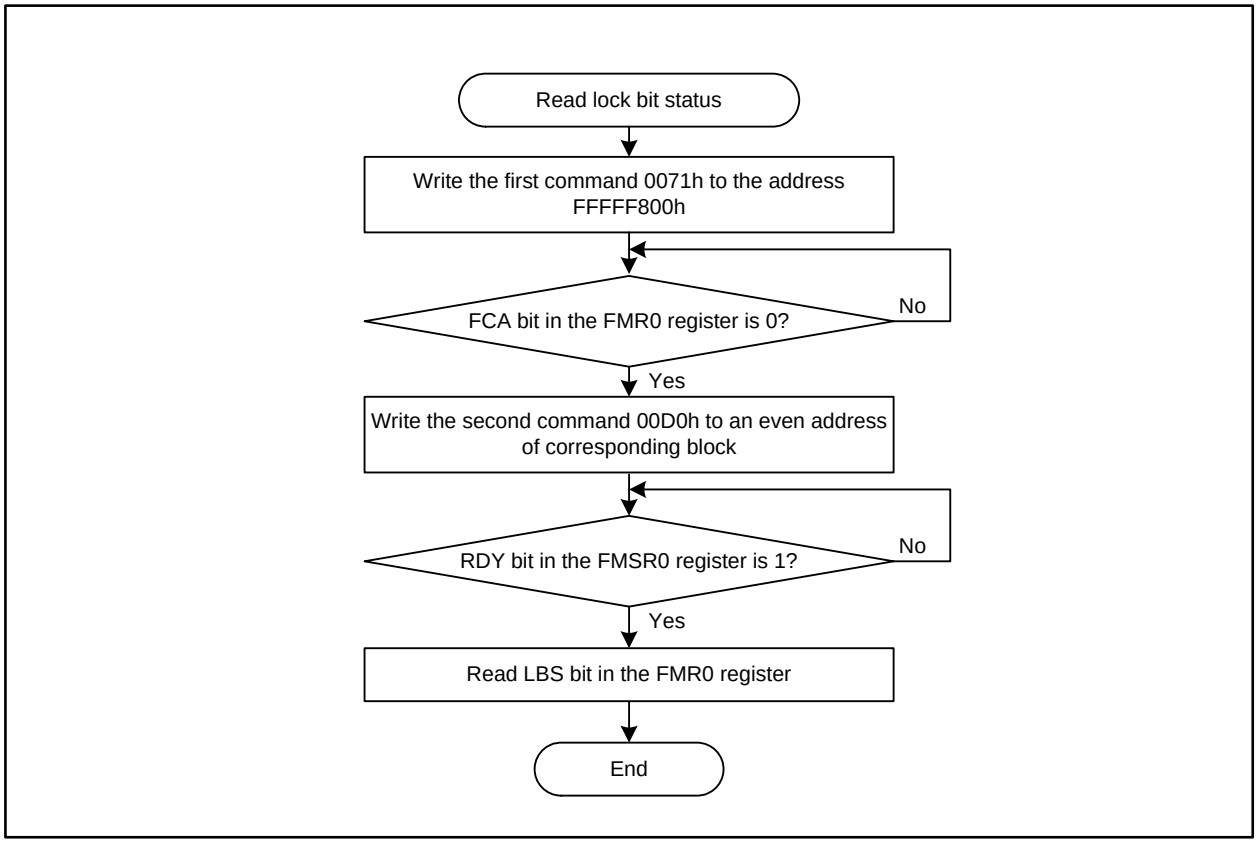

Read lock bit status
Write the first command 0071h to the address FFFFF800h
FCA bit in the FMR0 register is 0?
No
Yes
Write the second command 00D0h to an even address of corresponding block
RDY bit in the FMSR0 register is 1?
No
Yes
Read LBS bit in the FMR0 register
End

Figure 26.18 Read Lock Bit Status Command Flow

26.3.4.8 Enter Read Lock Bit Status Mode Command

This command is executed to enter read lock bit status mode. This command is enabled when the LBM bit in the FMR0 register is 0 (read via data bus).
To read the lock bit status of the read block, write 0071h to address FFFFF800h (Refer to Table 26.14).
The status is read in any address of the flash memory.
Execute this command in RAM.

26.3.4.9 Protect Bit Program Command

This command is executed to protect a block specified in the flash memory. ROM code protection is enabled by setting any protect bit of blocks to 0.

To program the protect bit of the designated block to 0 (protected), write 0067h to address FFFFF800h, verify that the FCA bit in the FMR0 register is 0, then write 00D0h to the protect bit of the corresponding block (Refer to Table 26.4).

To monitor the protect bit program, read the RDY bit in the FMSR0 register. This bit shows 0 (busy) when the operation is in progress and 1 (ready) when the operation is completed.

To verify the state of protect bit, enter read protect bit status mode (Refer to 26.3.4.10 “Enter Read Protect Bit Status Mode Command”), then read the flash memory.

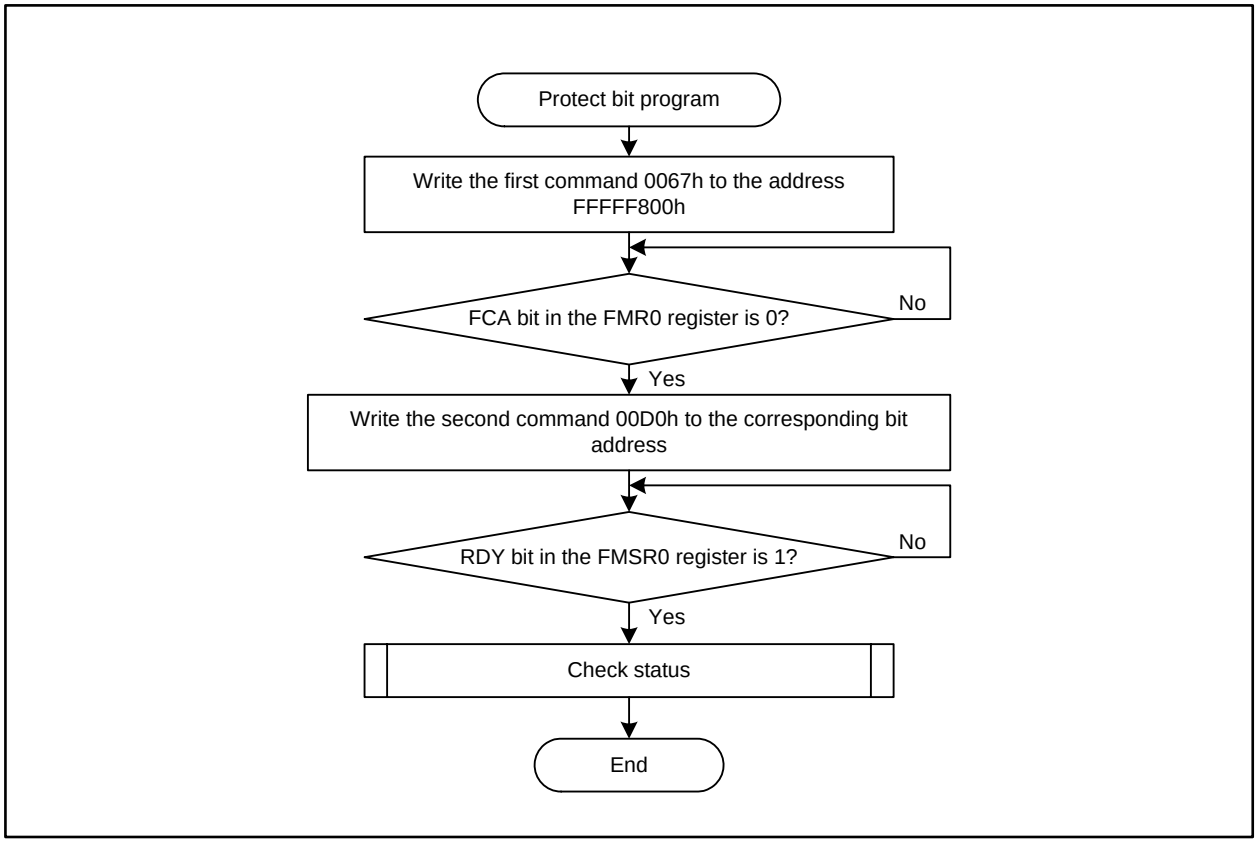

Protect bit program
Write the first command 0067h to the address FFFFF800h
FCA bit in the FMR0 register is 0?
No
Yes
Write the second command 00D0h to the corresponding bit address
RDY bit in the FMSR0 register is 1?
No
Yes
Check status
End

Figure 26.19 Protect Bit Program Command Flow

26.3.4.10 Enter Read Protect Bit Status Mode Command

This command is executed to enter read protect bit status mode.

To read the protect bit status of the read block, write 0061h to address FFFFF800h (Refer to Table 26.15). The status is read from any address of the flash memory.

Execute this command in RAM.

26.3.5 Status Check

To verify if a software command is successfully executed, read EERR or WERR bit in the FMSR0 register, or SR5 or SR4 bit in the status register.
Table 26.16 lists status and errors indicated by these bits and Figure 26.20 shows the flow of status check.

Table 26.16 Status and Errors

FMSR0 register (Status register)		Error	Causes for Error
EERR bit (SR5 bit)	WERR bit (SR4 bit)		
1	1	Command sequence error	• Data other than 00D0h or 00FFh (command to cancel) was written as the last command of two commands • An unavailable address was specified by an address specifying command
1	0	Erase error	• A locked block was tried to erase • Corresponding block was not erased properly
0	1	Program error	• A locked block was tried to program • Data was not programmed properly • Lock bit was not programmed properly • Protect bit was not programmed properly
0	0	No error	

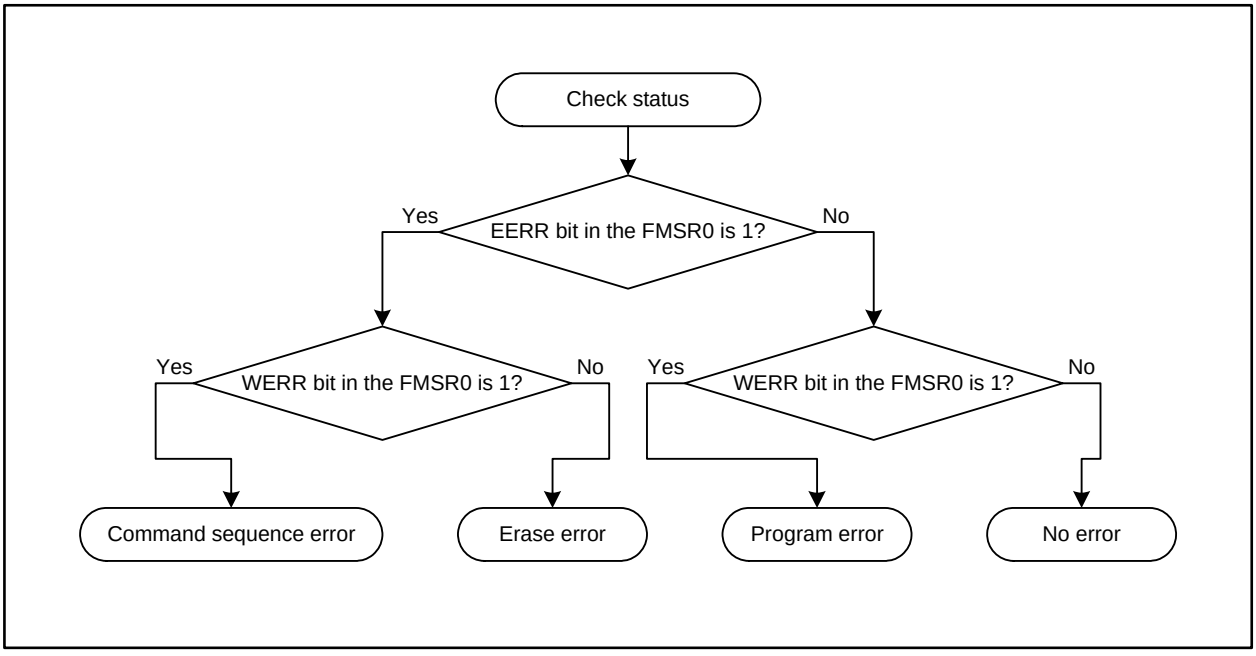


Figure 26.20 Status Check Flow

When an error occurs, execute clear status register command, then handle the error properly.
If erase errors or program errors occur frequently even though the program is correct, the corresponding block may be disabled.

26.4 Standard Serial I/O Mode

In standard serial I/O mode, the serial programmer supporting the R32C/102 Group can be used to rewrite the flash memory, while the MCU is mounted on a board.
For further information on the serial programmer, contact your serial programmer manufacturer and refer to the user's manual included with your serial programmer for instructions.
This mode provides two types of transmit/receive mode: Standard serial I/O mode 1 which uses synchronous serial interface and standard serial I/O mode 2 which uses UART as shown in Table 26.17.

Table 26.17 Standard Serial I/O Mode Specifications

Item		Standard Serial I/O Mode 1	Standard Serial I/O Mode 2
Transmit/receive mode		Synchronous serial I/O	UART
Transmit/receive bit rate		High	Low
Serial interface to be used		UART1	UART1
Pin setting	CNVSS	High	High
	$\overline{CE}$ (P5_0)	High	High
	$\overline{EPM}$ (P5_5)	Low	Low
	SCLK (P6_5)	In reset: Low In transmission/reception: Transmit/receive clock	In reset: Low In transmission/reception: Unused
Pin function	BUSY (P6_4)	BUSY signal	Monitor to check program operation
	RXD (P6_6)	Serial data input	Serial data input
	TXD (P6_7)	Serial data output	Serial data output

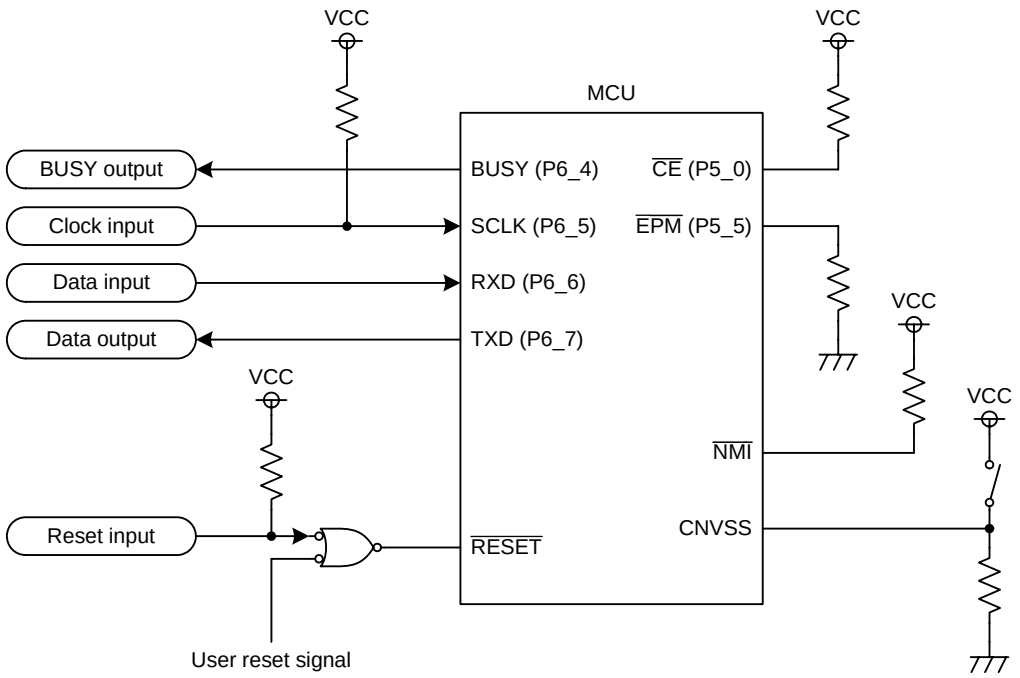
Table 26.18 and Table 26.19 list the pin definitions and functions in standard serial I/O mode. Figure 26.21 and Figure 26.22 show examples of a circuit application in standard serial I/O modes 1 and 2, respectively. Refer to the user's manual of your serial programmer to handle pins controlled by the serial programmer.

Table 26.18 Pin Definitions and Functions in Standard Serial I/O Mode (1/2)

Symbol	Function	I/O	Description
VCC, VSS	Power supply input	I	Applicable as follows: VCC = guaranteed voltage for program/ erase operation, VSS = 0 V
VDC1, VDC0	Connecting pins for decoupling capacitor	—	A decoupling capacitor for internal voltage should be connected between VDC0 and VDC1
CNVSS	CNVSS	I	This pin should be connected to VCC via a resistor
$\overline{\text{RESET}}$	Reset input	I	Reset input pin. While the $\overline{\text{RESET}}$ pin is driven low, a clock of 20 cycles or more should be input at the XIN pin
XIN	Main clock input	I	A ceramic resonator or a crystal oscillator should be connected between pins XIN and XOUT. An external clock should be input at the XIN while leaving the XOUT open
XOUT	Main clock output	O	
NSD	Debug port	I/O	This pin should be connected to VCC via a resistor of 1 to 4.7 kΩ
AVCC, AVSS	Analog power supply	I	AVCC and AVSS should be connected to VCC and VSS, respectively
VREF	Reference voltage input	I	Reference voltage input for the A/D converter and D/A converter
P0_0 to P0_7, P1_0 to P1_7, P2_0 to P2_7, P3_0 to P3_7, P4_0 to P4_7	Input port	I	High or low should be input, or the ports should be left open
P5_0	$\overline{\text{CE}}$ input	I	High should be input
P5_1 to P5_4	Input port	I	High or low should be input, or the ports should be left open
P5_5	$\overline{\text{EPM}}$ input	I	Low should be input
P5_6, P5_7, P6_0 to P6_3	Input port	I	High or low should be input, or the ports should be left open
P6_4	BUSY output	O	Standard serial I/O mode 1: BUSY output pin Standard serial I/O mode 2: Program operation monitor
P6_5	SCLK input	I	Standard serial I/O mode 1: Serial clock input pin Standard serial I/O mode 2: Low should be input
P6_6	Data input RXD	I	Serial data input pin
P6_7	Data output TXD	O	Serial data output pin
P7_0 to P7_7, P8_0 to P8_4	Input port	I	High or low should be input, or the ports should be left open
P8_5	$\overline{\text{NMI}}$ input	I	This pin should be connected to VCC via a resistor
P8_6, P8_7, P9_0 to P9_7, P10_0 to P10_7, P11_3, P11_4, P12_0 to P12_7, P13_0 to P13_7, P14_1, P14_3 to P14_6, P15_0 to P15_3	Input port	I	High or low should be input, or the ports should be left open

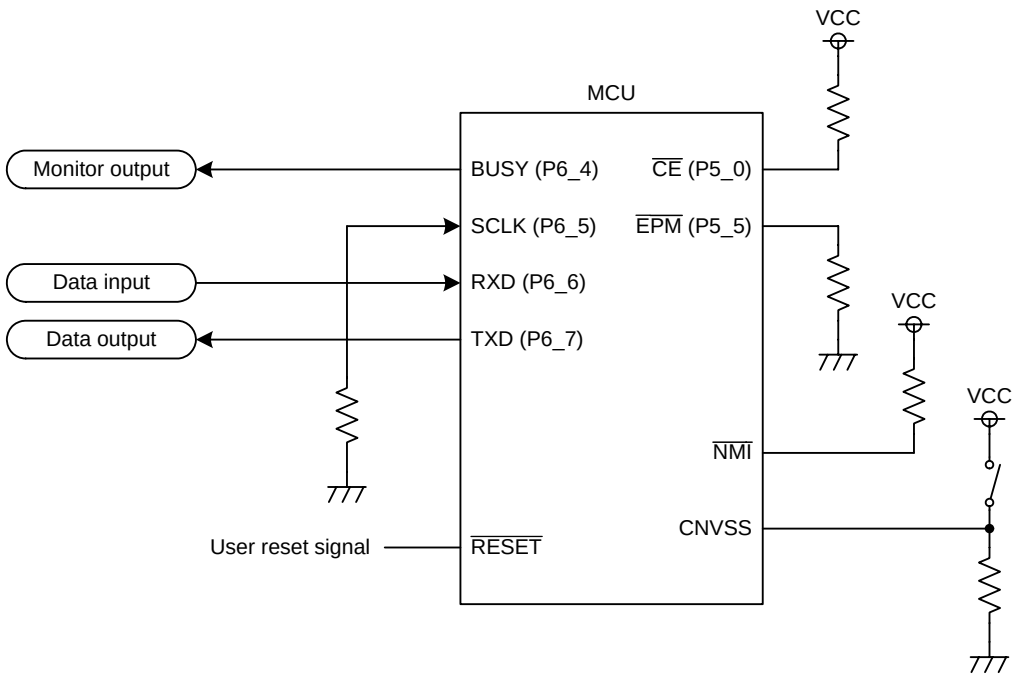
Table 26.19 Pin Definitions and Functions in Standard Serial I/O Mode (2/2)

Symbol	Function	I/O	Power Supply	Description
LVCC, LVSS	Power supply input	I	—	Applicable as follows: LVCC = 5 V, LVSS = 0 V
VSUB	Power supply input	I	—	Applicable as follows: VSUB = 0 V
VDC2	Connecting pin for decoupling capacitor	—	—	A decoupling capacitor for internal voltage should be connected between VDC2 and VSS
C_$\overline{\text{RESET}}$	Reset input	I	—	Low should be input
CVCC1 to CVCC3 CVSS1 to CVSS3	Power supply input	I	—	Applicable as follows: CVCC1 to CVCC3 = 5 V, CVSS1 to CVSS3 = 0 V
VCOMAFE, VCOMABE	Connecting pins for decoupling capacitor	—	—	A ceramic capacitor (0.1 µF approx.) and an electrolytic capacitor (100 µF approx.) should be connected in parallel between VCOMAFE and CVSS1, and between VCOMABE and CVSS2
SDIN0, SDIN1, LRCLK, BITCLK, MCLK	Audio interface	I	VCC	High or low should be input
SDOUT0 to SDOUT2	Audio interface	O	VCC	The pins should be left open
AINL, AINR	Analog input	I	CVCC1	The pins should be left open
AOUT0L to AOUT2L, AOUT0R to AOUT2R	Analog output	O	CVCC3	The pins should be left open



Notes:
1. Control pins and external circuitry vary with the serial programmer. Refer to the user's manual included with the serial programmer.
2. In this example, a selector controls the voltage applied to the CNVSS pin to switch between in single-chip mode and in standard serial I/O mode 1.
3. If the user reset signal becomes low while the MCU is communicating with the serial programmer, the connection between the user reset signal and the $\overline{\text{RESET}}$ pin by, for example, a jumper selector.

Figure 26.21 Circuit Application in Standard Serial I/O Mode 1



Notes:
1. Control pins and external circuitry vary with the serial programmer. Refer to the user's manual included with the serial programmer.
2. In this example, a selector controls the voltage applied to the CNVSS pin to switch between in single-chip mode and in standard serial I/O mode 2.
3. If the user reset signal becomes low while the MCU is communicating with the serial programmer, the connection between the user reset signal and the $\overline{\text{RESET}}$ pin by, for example, a jumper selector.

Figure 26.22 Circuit Application in Standard Serial I/O Mode 2

26.5 Parallel I/O mode

In parallel I/O mode, the parallel programmer supporting the R32C/102 Group can be used to rewrite the flash memory.

For further information on the parallel programmer, contact your parallel programmer manufacturer and refer to the user's manual included with your parallel programmer for instructions.

26.6 Notes on Flash Memory Rewriting

26.6.1 Note on Power Supply

- Keep the supply voltage constant within the range specified in the electrical characteristics while a rewrite operation on flash memory is in progress. If the supply voltage becomes beyond the guaranteed value, the device cannot be guaranteed.

26.6.2 Note on Hardware Reset

- Do not perform a hardware reset while a rewrite operation on flash memory is in progress.

26.6.3 Note on Flash Memory Protection

- If an ID code written in an assigned address has an error, any read/write operation of flash memory in standard serial I/O mode is disabled.

26.6.4 Notes on Programming

- Do not set the FEW bit in the FMCR register to 1 (CPU rewrite mode) in low speed mode or low power mode.
- Four software commands of program, block erase, lock bit program, and protect bit program are interrupted by an NMI, a watchdog timer interrupt, or an oscillator stop detection interrupt. If any of the software commands above is interrupted, erase the corresponding block and then execute the same command again. If the block erase command is interrupted, values of lock bits and protect bits become undefined. Therefore, disable the lock bit, and then execute the block erase command again.

26.6.5 Notes on Interrupts

- EW0 mode
  - To use interrupts assigned to the relocatable vector table, the vector table should be addressed in RAM space.
  - If either of an NMI, a watchdog timer interrupt, or an oscillator stop detection interrupt is generated, the flash memory module automatically enters read array mode. Therefore these interrupts are enabled even during a rewrite operation. On the other hand, the rewrite operation in progress is aborted by the interrupt and registers FMR0 and FRSR0 are reset. When the interrupt handler has ended, set the LBD bit in the FMR1 register to 1 (lock bit protection disabled) to re-execute the rewrite operation.
  - Instructions BRK, INTO, and UND, which refer to data on the flash memory, are unavailable in this mode.

- EW1 mode
  - Interrupts assigned to the relocatable vector table should not be accepted during a program or block erase operation.
  - The watchdog timer interrupt should not be generated, either.
  - If either of an NMI, a watchdog timer interrupt, or an oscillator stop detection interrupt is generated, the flash memory module automatically enters read array mode. Therefore this interrupt is enabled even during a rewrite operation. On the other hand, the rewrite operation in progress is aborted by the interrupt and registers FMR0 and FRSR0 are reset. When the interrupt handler has ended, set the EWM bit in the FMR0 register to 1 (set as EW1 mode) and the LBD bit in the FMR1 register to 1 (lock bit protection disabled) to re-execute the rewrite operation.

26.6.6 Notes on Rewrite Control Program

- EW0 mode
  - If the supply voltage lowers during the rewrite operation of blocks having the rewrite control program, the rewrite control program may not be successfully rewritten, then the rewrite operation itself may not be performed. In this case perform the rewrite operation by serial programmer or parallel programmer.

- EW1 mode
  - Do not rewrite blocks having the rewrite control program.

26.6.7 Notes on Number of Programming/Erasure and Software Command Execution Time

- According to the increase of program/erase operation, the four software commands: program, block erase, lock bit program, and protect bit program require more time to be executed. If the number of programming/erasure exceeds the minimum endurance value specified in the electrical characteristics, it may take unpredictable time to execute the software commands. The waiting time for the execution of software commands should be set much longer than the execution time specified in the electrical characteristics.

26.6.8 Other Notes

- The required time to perform the program or erase operation specified in the electrical characteristics can be guaranteed within the minimum values of programming/erasure endurance specified in the same table. Even if the number of programming/erasure exceeds the minimum endurance value, the program or erase operation may be unguaranteedly performed.
- Chips repeatedly programmed and erased for debugging are not allowed to be used for commercial products.

27. Digital Audio Processor (DAP)

The R32C/102 Group supports a digital audio processor (DAP) comprised of two channels of 24-bit A/D converter, an audio DSP, and six channels of 24-bit D/A converter. Table 27.1 lists the specifications and Figure 27.1 shows a block diagram of DAP.

Table 27.1 DAP Specifications

Item		Specification
Audio DSP		CoolFlux DSP of NXP Semiconductors is embedded • Dual Harvard memory architecture • Multiply-accumulate unit: 24-bit × 24-bit + 56-bit →56-bit (2 circuits) • DSP performance: 1536 step/fs or 1024 step/fs • Program RAM: 12 Kbytes (1536 step/fs)/6 Kbytes (1024 step/fs) • Data RAM: 36 Kbytes (1536 step/fs)/12 Kbytes (1024 step/fs) • Coefficient RAM: 5376 bytes (1536 step/fs)/1536 bytes (1024 step/fs)
Clock		DSP clock is generated by PLL with MCLK input from external oscillator • MCLK frequency: 256 fs or 384 fs • PLL synthesizer: multiplied by 2.667/ by 4 or by 6
Serial Interface		Synchronous serial interface × 1 channel
Parallel Interface		8-bit general purpose I/O port × 1
Audio Interface		Serial audio interface (SAI) × 5 channels • Master/slave selectable
	Data format	MSB first & left-justified, IIS, MSB first & right-justified
	Data size	16-, 20-, or 24-bit word length
	BITCLK	64 fs or 48 fs
	Input	2 channels
	Output	3 channels
Control Interface		Synchronous serial interface with $\overline{WE}$ × 1 channel
A/D Converter		24-bit resolution × 2 channels • Single-ended input
D/A Converter		24-bit resolution × 6 channels • Single-ended output

The DAP, provided as an MCU-independent module, is controlled via ports P11_0 to P11_2 and ports P15_4 to P15_7. The former ports are used as synchronous serial interface to communicate with DSP and the latter are used to control the module itself. Note that the ports P15_4 and P15_6 should be used as synchronous serial interface or as general ports for serial communication by a program. When these ports are set for the function of UART6, the same function in both ports P4 and P12 is disabled.
DAP general purpose I/O ports DP_0 to DP_7 are connected to MCU ports P10_0 to P10_7 in a wired-OR configuration. These ports enable communication between the MCU and DSP.

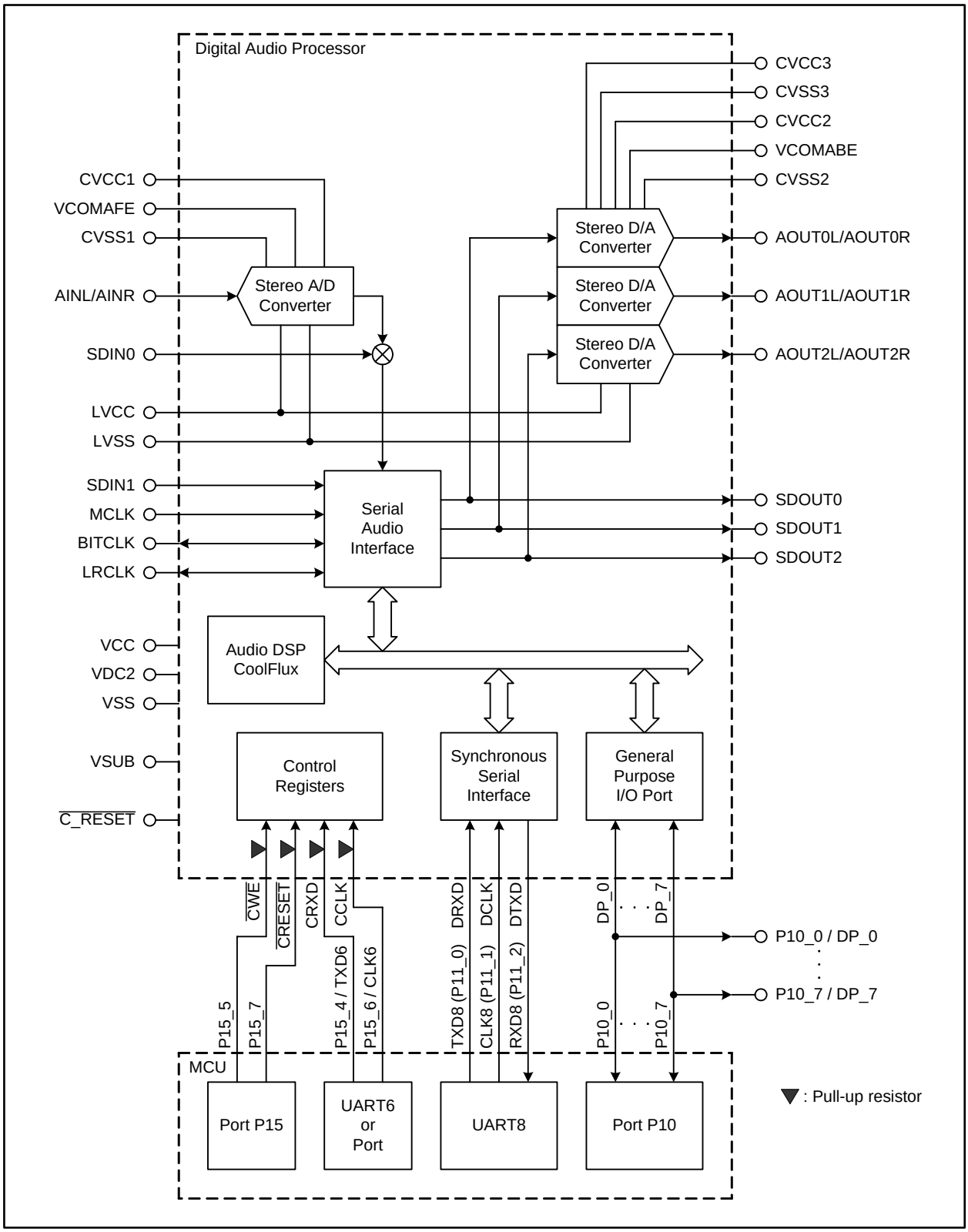


Figure 27.1 DAP Block Diagram

27.1 A/D Converter

The DAP incorporates a 24-bit delta-sigma stereo A/D converter. Table 27.2 lists the specifications and Figure 27.2 shows a block diagram of the A/D converter.

Table 27.2 A/D Converter Specifications

Item		Specification
Conversion Structure		Fifth-order, 64fs, 1-bit delta-sigma modulator
Resolution		24 bits
Sampling Frequency		44.1 kHz and 48 kHz
System Clock Frequency		256 fs or 384 fs
Decimation Filter	Type	1/64-times-digital filter
	Pass band	Up to 0.454 fs (20 kHz@fs = 44.1 kHz, 21.8 kHz@fs = 48 kHz)
	Stop band	0.546 fs and over (24 kHz@fs = 44.1 kHz, 26.2 kHz@fs = 48 kHz)
	Pass-band ripple	±0.005 dB
	Stop-band attenuation	-80 dB
DC Cut Filter	Type	High-pass filter
	Cutoff frequency	1.0 Hz (-3 dB)
	-1.0 dB frequency	2.9 Hz
	-0.1 dB frequency	6.5 Hz
Input		Single-ended
Input Impedance		33 kΩ (typ.)
Input Bias Voltage		½ CVCC1 (typ.)
Full Scale Maximum Input Voltage		1.1 V_{RMS} (typ.)
Signal-to-noise Ratio		100 dB (typ.)
Total Harmonic Distortion-plus-Noise to Signal Ratio		92 dB (typ.)
Dynamic Range		100 dB (typ.)
Inter-channel Crosstalk		-105 dB (typ.)
Inter-channel Level Difference		±0.5 dB
Power Supply	Supply voltage	+5.0 V (CVCC1), 0 V (CVSS1)
	Referential voltage	+2.5 V (VCOMAFE)
	Supply voltage for level shifter	+5.0 V (LVCC), 0 V (LVSS)
	Logic supply voltage	+3.3 V (VCC), 0 V (VSS)

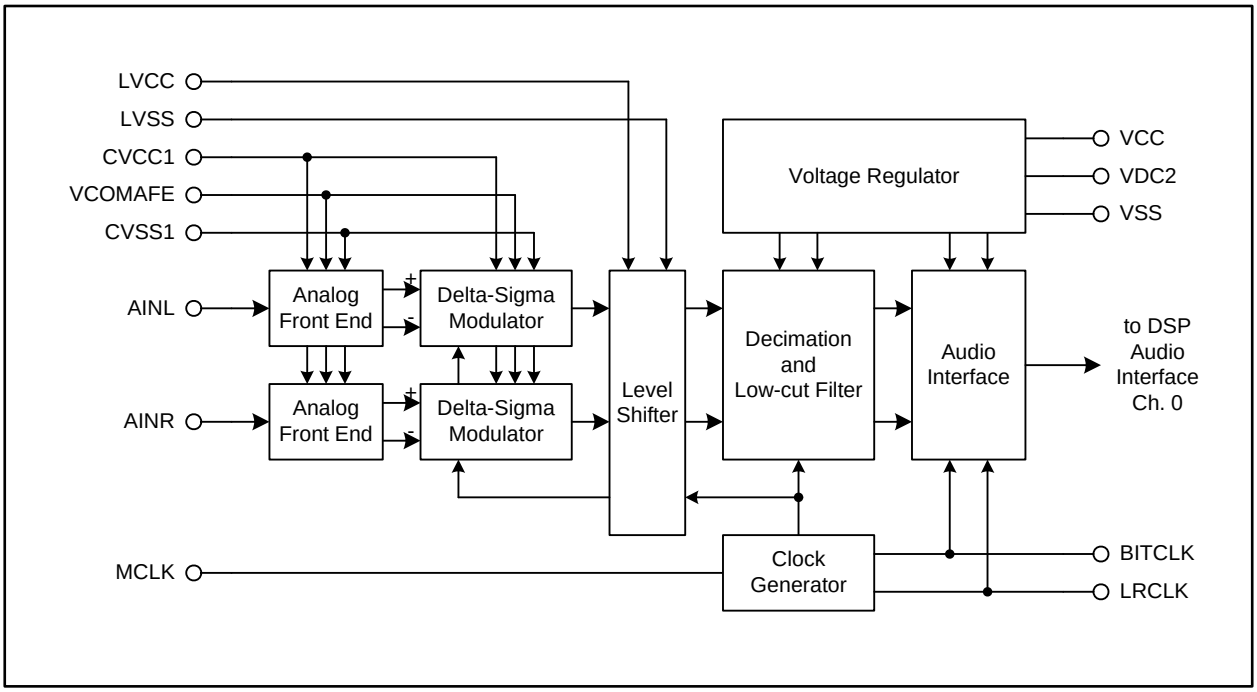


Figure 27.2 Block Diagram of A/D Converter

27.2 D/A Converter

The DAP incorporates three 24-bit delta-sigma stereo D/A converters. Table 27.3 lists the specifications and Figure 27.3 shows a block diagram of the D/A converter.

Table 27.3 D/A Converter Specifications

Item		Specification
Conversion Structure		Third-order, 64fs, 16-element delta-sigma modulator
Resolution		24 bits
Sampling Frequency		44.1 kHz or 48 kHz
System Clock Frequency		256 fs or 384 fs
Gap-filling Filter	Type	8-times oversampling digital filter
	Pass band	Up to 0.454 fs (20 kHz@fs = 44.1 kHz, 21.8 kHz@fs = 48 kHz)
	Stop band	0.546fs and over (24 kHz@fs = 44.1 kHz, 26.2 kHz@fs = 48 kHz)
	Pass-band ripple	±000.5 dB (max.)
	Stop-band attenuation	-80 dB
Low-pass Filter	Filter order	First order (-6 dB per octave)
	Cutoff frequency	177 kHz
Output		Single-ended
Output Impedance		110 kΩ (typ.)
Output Bias Voltage		½ CVCC3 (typ.)
Maximum Output Voltage		1.1 Vrms (typ.)
Signal-to-noise Ratio		100 dB (typ.)
Total Harmonic Distortion-plus-Noise to Signal Ratio		90 dB (typ.)
Dynamic Range		100 dB (typ.)
Inter-channel Crosstalk		-110 dB (typ.)
Inter-channel Level Difference		±0.5 dB (max.)
Power Supply	Supply voltage	+5.0 V (CVCC2, CVCC3), 0 V (CVSS2, CVSS3)
	Referential voltage	+2.5 V (VCOMABE)
	Supply voltage for level shifter	+5.0 V (LVCC), 0 V (LVSS)
	Logic supply voltage	+3.3 V (VCC), 0 V (VSS)

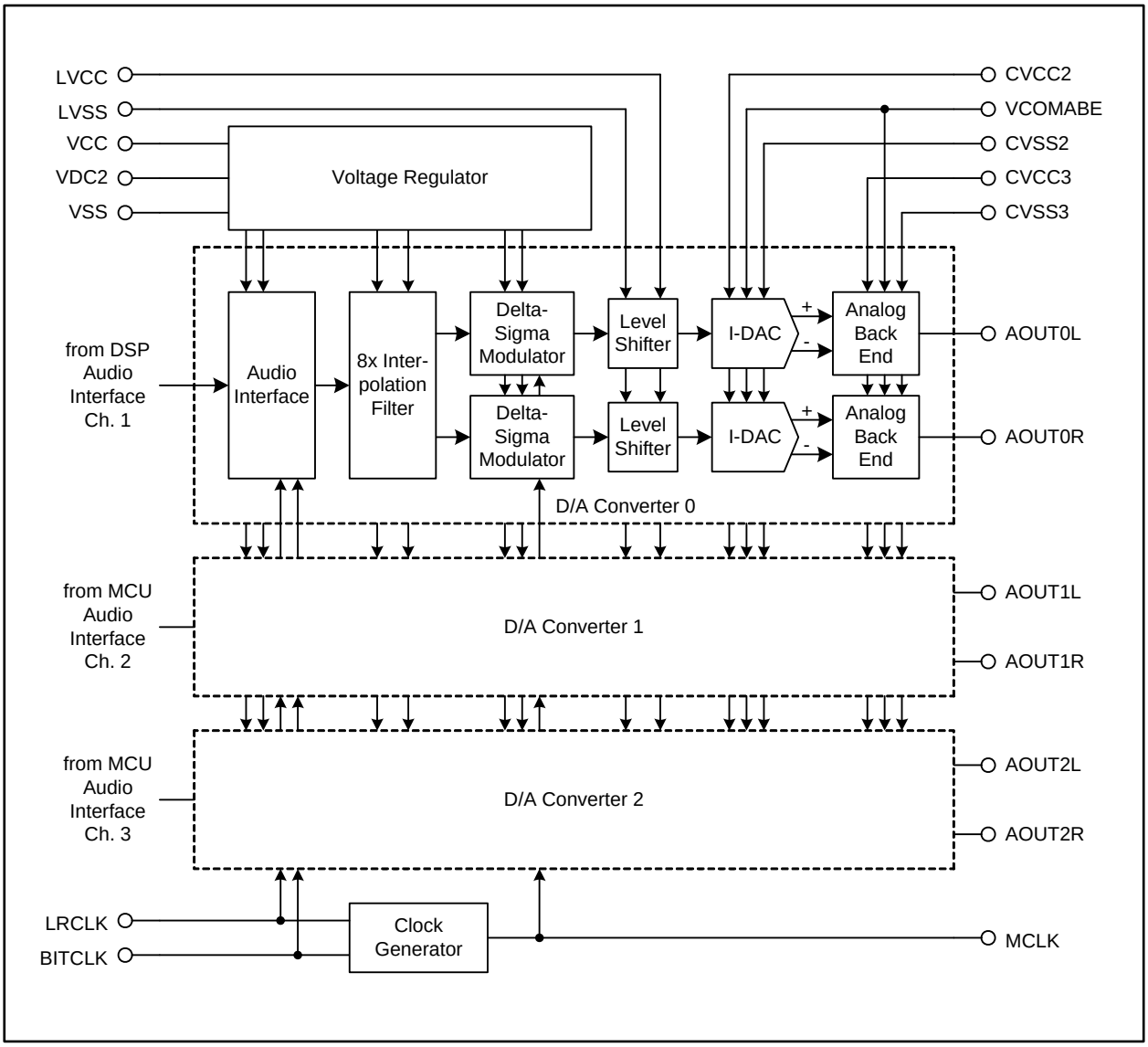


Figure 27.3 Block Diagram of D/A Converter

27.3 DAP Control

The DAP is controlled by the C_RESET pin and ports P15_4 to P15_7.

27.3.1 Reset

To reset the DAP, the C_RESET pin or port P15_7 is used. The C_RESET pin should be connected to the RESET pin.

The DAP is reset when a low signal is applied to the C_RESET pin under the recommended operating conditions of supply voltage. It is also reset when a low signal is output from the port P15_7 of MCU.

Note that the reset is not released although the reset signal mentioned above is driven high. The reset state is released by a control command.

27.3.2 Control Protocol

Control registers of the DAP are not mapped on SFR, which disables the CPU to directly control the registers by read/write access. The DAP is controlled by transmitting a control command in serial transmission using ports P15_4 to P15_6.

The control command is issued by transmitting 8-bit serial data twice as shown in Figure 27.4. The first data specifies the address of a register, and the second data specifies data to be controlled. The data is transmitted in 8-bit synchronous serial interface mode from the LSB. The port P15_5 (CWE signal) should be held low during transmission. The data is written to the internal register on the rising edge of the CWE signal.

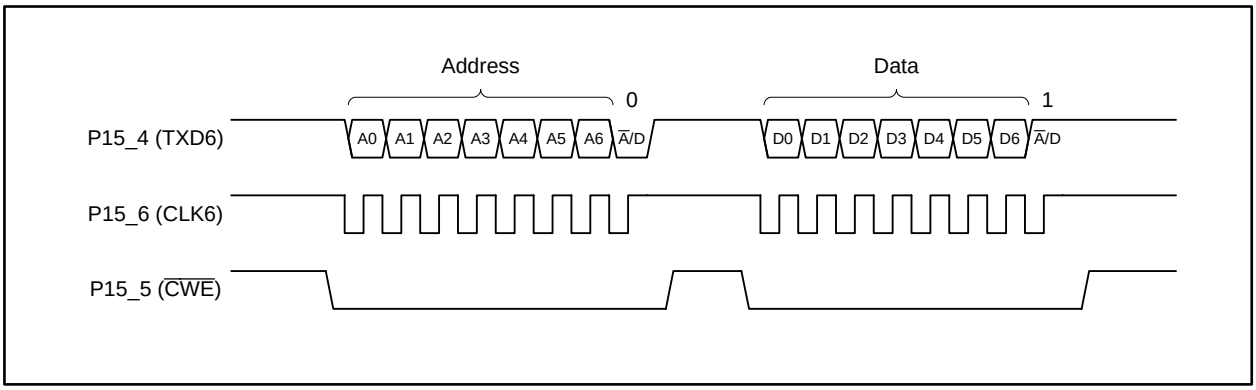


Figure 27.4 Control Command Format

27.4 DAP Control Registers

The DAP has 18 control registers as listed in Table 27.4.

Table 27.4 DAP Control Registers

Address	Register	Symbol	Reset Value
01h	CODEC control register	CDCR	0000 0000b
02h	Audio interface configuration register	AIFC	0000 0000b
03h	DAP PLL configuration register	DPLC	0010 0100b
04h	DSP control register	DSPC	0000 0000b
05h	DAP voltage regulator control register	DVRC	0000 0000b
06h	CODEC data selector control register	CDSC	0000 0000b
08h	CODEC A/D converter dither control register	CADDS	0000 0000b
09h	CODEC D/A converter dither control register	CDADS	0000 0000b
10h	DAP I/O pin control register 0	DPC0	0000 0000b
11h	DAP I/O pin control register 1	DPC1	0000 0000b
12h	DAP I/O pin control register 2	DPC2	0000 0000b
13h	DAP I/O pin control register 3	DPC3	0000 0000b
14h	DAP I/O pin control register 4	DPC4	0000 0000b
15h	DAP I/O pin control register 5	DPC5	0000 0000b
16h	DAP I/O pin control register 6	DPC6	0000 0000b
17h	DAP I/O pin control register 7	DPC7	0000 0000b
18h	DAP I/O pin control register 8	DPC8	0000 0000b
1Ah	DAP I/O pin control register 10	DPC10	0000 0000b
1Ch	DAP input signal control register 0	DIC0	0000 0000b
1Dh	DAP input signal control register 1	DIC1	0000 0000b
1Fh	DAP protect register	DPRR	0000 0000b
2Dh	DAP clock control register	DCKC	0000 0000b

The following pages describes the details on these registers. Note that each figure shows eight bits in total: seven enable bits for each control register and one bit to be set to 1 as the $\overline{A}$/D bit of serial data.

27.4.1 CODEC Control Register (CDCR Register)

CODEC Control Register

b7	b6	b5	b4	b3	b2	b1	b0
1	0	0					

Symbol: CDCR
Address: 01h
Reset Value: 0000 0000b

Bit Symbol	Bit Name	Function	RW
CRR	CODEC Reset Release Bit	0: Reset 1: Reset released	WO
PCAD	A/D Converter Power Control Bit	0: Power-off 1: Power-on	WO
PCDA0	D/A Converter 0 Power Control Bit	0: Power-off 1: Power-on	WO
PCDA1	D/A Converter 1 Power Control Bit	0: Power-off 1: Power-on	WO
PCDA2	D/A Converter 2 Power Control Bit	0: Power-off 1: Power-on	WO
— (b6-b5)	Reserved	Should be written with 0	WO
— (b7)	Reserved	Should be written with 1	WO

Figure 27.5 CDCR Register

CRR Bit

An reset of CODEC block, specifically an A/D converter, D/A converters, an audio interface, and the PLL frequency synthesizer, is released when this bit is set to 1 after the main regulator provides output power stably.

When this bit is set to 0, each module of the CODEC is reset. The MCLK is also stopped. To stop the main regulator, set this bit to 0 before setting the MRC bit in the DVRC register to 0 (main regulator stopped).

PCAD Bit

This bit controls the power supply of A/D converter. When this bit is 1, the supply voltage is applied to the A/D converter. When this bit is 0, the power supply is stopped.

When not using the A/D converter, the power can be saved by setting this bit to 0.

Bits PCDA0 to PCDA2

These bits controls the power supply of D/A converter. When these bits are 1, the supply voltage is applied to the respective D/A converters. When these bits are 0, the power supply is stopped.

When not using the D/A converter, the power can be saved by setting this bit to 0.

27.4.2 Audio Interface Configuration Register (AIFC Register)

Audio Interface Configuration Register (1)

b7	b6	b5	b4	b3	b2	b1	b0
1							

Symbol	Address	Reset Value
AIFC	02h	0000 0000b

Bit Symbol	Bit Name	Function	RW
MCKS	MCLK Select Bit	0: 256 fs (2) 1: 384 fs	WO
BCKS	BITCLK Select Bit (2, 3)	0: 64 fs 1: 48 fs	WO
MSS	Master/Slave Select Bit	0: Master (BITCLK and LRCLK output internal clock) 1: Slave (BITCLK and LRCLK input external clock)	WO
FMT0	Data Format Select Bit	Data format and data word length are: b6 b5 b4 b3 0 0 0 0 : Left-justified, 16 bits 0 0 0 1 : Left-justified, 20 bits 0 0 1 0 : Left-justified, 24 bits 0 1 0 0 : IIS, 16 bits 0 1 0 1 : IIS, 20 bits 1 0 1 0 : IIS, 24 bits (3) 1 1 0 0 : Right-justified, 16 bits (3) 1 1 0 1 : Right-justified, 20 bits (3) 1 1 1 0 : Right-justified, 24 bits (3) Only use the combinations listed above	WO
FMT1			WO
FMT2			WO
FMT3			WO
— (b7)	Reserved	Should be written with 1	WO

Notes:
1. This register should be rewritten when all the following bits in the CDCR register are 0 (power-off): PCAD, PCDA0, PCDA1, and PCDA2.
2. Set the BCKS bit to 0 when the MCKS bit is 0.
3. Set the BCKS bit to 0 when using the 24-bit IIS format or a right-justified format.

Figure 27.6 AIFC Register

MCKS Bit

This bit selects a frequency for MCLK. A clock which is 1 fs of frequency multiplexed by 256 or 384 is provided from the MCLK pin.

BCKS Bit

This bit selects the frequency for BITCLK. When using 256 fs of frequency for MCLK, use 64 fs of frequency for BITCLK.

MSS Bit

This bit selects BITCLK and LRCLK from internally generated (master) or externally provided (slave).

Bits FMT3 to FMT0

This bits selects a format for serial audio interface.

27.4.3 DAP PLL Configuration Register (DPLC Register)

DAP PLL Configuration Register (1)

b7	b6	b5	b4	b3	b2	b1	b0
1				0	1		0

Symbol: DPLC
Address: 03h
Reset Value: 0010 0100b

Bit Symbol	Bit Name	Function	RW
— (b0)	Reserved	Should be written with 0	WO
MCV	Main Counter Divide Ratio Setting Bit (2)	0: Divide-by-16 1: Divide-by-24	WO
— (b2)	Reserved	Should be written with 1	WO
— (b3)	Reserved	Should be written with 0	WO
RCV0	Reference Counter Divide Ratio Setting Bit (2)	b5 b4 0 1: Divide-by-4 1 0: Divide-by-6 Only use the combinations listed above	WO
RCV1			WO
OSC	PLL Oscillator Control Bit	0: Start PLL oscillator running 1: Stop PLL oscillator running	WO
— (b7)	Reserved	Should be written with 1	WO

Notes:
1. Set the WE bit in the DPRR register to 1 (write enabled) before rewriting this register.
2. Set these bits when the OSC bit is 0. The values are rewritable only once.

Figure 27.7 DPLC Register

Bits MCV, RCV1 and RCV0
The PLL multiplication ratio is specified by the combination of these bits. Refer to the table below for the PLL configuration.

Table 27.5 PLL Configuration

RCV1	RCV0	MCV	PLL Multiplication Ratio	MCKS	MCLK	PLL
0	1	0	1/4 × 16 = 4	0	256 fs	1024 fs
				1	384 fs	1536 fs
0	1	1	1/4 × 24 = 6	0	256 fs	1536 fs
				1	384 fs	Do not set this combination
1	0	0	1/6 × 16 = 2.67	0	256 fs	Do not set this combination
				1	384 fs	1024 fs
1	0	1	1/6 × 24 = 4	0	256 fs	1024 fs
				1	384 fs	1536 fs

OSC Bit
When this bit is set to 1, the PLL starts oscillating. Before setting this bit to 1, set bits MCV, RCV1, and RCV0. To stop the main regulator, set this bit to 0 before the MRC bit in the DVRC register to 0 (main regulator stopped).

27.4.4 DSP Control Register (DSPC Register)

DSP Control Register

b7	b6	b5	b4	b3	b2	b1	b0
1	0	0	0	0	0		

Symbol: DSPC
Address: 04h
Reset Value: 0000 0000b

Bit Symbol	Bit Name	Function	RW
DRR	DSP Reset Release Bit	0: Reset 1: Reset released	WO
DMOD	DSP Mode Select Bit	0: Program transfer mode 1: Program execute mode	WO
— (b6-b2)	Reserved	Should be written with 0	WO
— (b7)	Reserved	Should be written with 1	WO

Figure 27.8 DSPC Register

DRR Bit

An reset of the DSP is released when this bit is set to 1 after the PLL frequency synthesizer provides oscillation stably.

When this bit is set to 0, the DSP is reset. To stop the main regulator, set this bit to 0 before set the MRC bit in the DVRC register to 0 (main regulator stopped).

DMOD Bit

This bit selects an operation mode of DSP.

When the DSP is released from reset state while this bit is 0, a DSP bootloader program (1) starts running to transfer program and data to DSP internal memory.

When the DSP is released from reset state while this bit is 1, program on DSP internal program memory is executed.

Note:

1. Please contact a Renesas Electronics sales office for the details on the bootloader program.

27.4.5 DAP Voltage Regulator Control Register (DVRC Register)

DAP Voltage Regulator Control Register (1)

b7	b6	b5	b4	b3	b2	b1	b0
1	0	0	0	0	0	0	

Symbol	Address	Reset Value
DVRC	05h	0000 0000b

Bit Symbol	Bit Name	Function	RW
MRC	Main Regulator Control Bit	0: Main regulator stopped 1: Main regulator active	WO
— (b6-b1)	Reserved	Should be written with 0	WO
— (b7)	Reserved	Should be written with 1	WO

Note:
1. Set the WE bit in the DPRR register to 1 (write enabled) before rewriting this register.

Figure 27.9 DVRC Register

MRC Bit

This bit controls the main regulator which generates the supply voltage for internal logic of DAP. When this bit is set to 0, the main regulator stops and the power supply to any circuits other than those of control registers is stopped. When not using DAP, the power can be saved by setting this bit to 0. When this bit is set to 1, the main regulator starts running, the supply voltage is applied to each module. Note that a certain waiting time is required until the supply voltage for internal logic has stabilized.

27.4.6 CODEC Data Selector Control Register (CDSC Register)

CODEC Data Selector Control Register

b7	b6	b5	b4	b3	b2	b1	b0
1	0	0	0	0	0	0	

Symbol: CDSC
Address: 06h
Reset Value: 0000 0000b

Bit Symbol	Bit Name	Function	RW
DSPIS	DSP Input Data Select Bit	0: On-chip A/D converter output 1: Input from SDIN0 pin	WO
— (b6-b1)	Reserved	Should be written with 0	WO
— (b7)	Reserved	Should be written with 1	WO

Figure 27.10 CDSC Register

DSPDI Bit

This bit is used to switch input signal for data to be processed in the DSP.
When this bit is set to 0, serial audio data from the on-chip A/D converter is used.
When this bit is set to 1, serial audio data from the SDIN0 pin is used.

27.4.7 CODEC A/D Converter Dither Control Register (CADDS Register)

CODEC A/D Converter Dither Control Register

b7							b0
1	0	0					

Symbol: CADDS
Address: 08h
Reset Value: 0000 0000b

Bit Symbol	Bit Name	Function	RW
DCL	Left Channel Dither Control Bit	0: Dither-off 1: Dither-on	WO
DAL	Left Channel Dither Amount Select Bit	0: High 1: Low	WO
DCR	Right Channel Dither Control Bit	0: Dither-off 1: Dither-on	WO
DAR	Right Channel Dither Amount Select Bit	0: High 1: Low	WO
DWF	Dither Waveform Select Bit	0: 25% duty cycle of square wave 1: 50% duty cycle of square wave	WO
— (b6-b5)	Reserved	Should be written with 0	WO
— (b7)	Reserved	Should be written with 1	WO

Figure 27.11 CADDS Register

This register is used to control the dither amount to be applied to the A/D converter. The recommended setting is “00010101b”.

27.4.8 CODEC D/A Converter Dither Control Register (CDADS Register)

CODEC D/A Converter Dither Control Register

b7	b6	b5	b4	b3	b2	b1	b0
1	0	0	0	0	0		

Symbol	Address	Reset Value
CDADS	09h	0000 0000b

Bit Symbol	Bit Name	Function	RW
— (b1-b0)	—	Set the amount of dither to add to D/A converter input	WO
— (b6-b2)	Reserved	Should be written with 0	WO
— (b7)	Reserved	Should be written with 1	WO

Figure 27.12 CDADS Register

This register is used to control the dither amount to be applied to the D/A converter. The recommended setting is “00000001b”.

27.4.9 DAP I/O Pin Control Register 0 (DPC0 Register)

DAP I/O Pin Control Register 0

b7	b6	b5	b4	b3	b2	b1	b0
1	1		1	1	1		0

Symbol	Address	Reset Value
DPC0	10h	0000 0000b

Bit Symbol	Bit Name	Function	RW
— (b0)	Reserved	Should be written with 0	WO
SDI0D	SDIN0 Pin I/O Select Bit	0: Input 1: Fixed low level output	WO
— (b4-b2)	Reserved	Should be written with 1	WO
SDI1D	SDIN1 Pin I/O Select Bit	0: Input 1: Fixed low level output	WO
— (b7)	Reserved	Should be written with 1	WO

Figure 27.13 DPC0 Register

27.4.10 DAP I/O Pin Control Register 1 (DPC1 Register)

DAP I/O Pin Control Register 1

b7	b6	b5	b4	b3	b2	b1	b0
1	0	0	0	0	1		

Symbol	Address	Reset Value
DPC1	11h	0000 0000b

Bit Symbol	Bit Name	Function	RW
BCKD	BITCLK Pin I/O Select Bit (1)	0: Input 1: Output	WO
LRCD	LRCLK Pin I/O Select Bit (1)	0: Input 1: Output	WO
— (b2)	Reserved	Should be written with 1	WO
— (b6-b3)	Reserved	Should be written with 0	WO
— (b7)	Reserved	Should be written with 1	WO

Note:
1. When The MSS bit in the AIFC register is 0 (master), set this bit to 1. When the MSS bit is 1 (slave), set this bit to 0.

Figure 27.14 DPC1 Register

27.4.11 DAP I/O Pin Control Register 2 (DPC2 Register)

DAP I/O Pin Control Register 2

b7	b6	b5	b4	b3	b2	b1	b0
1	0						

Symbol	Address	Reset Value
DPC2	12h	0000 0000b

Bit Symbol	Bit Name	Function	RW
DP0D	DP_0 Pin I/O Select Bit	0: Input 1: Output (1)	WO
DP1D	DP_1 Pin I/O Select Bit	0: Input 1: Output (1)	WO
DP2D	DP_2 Pin I/O Select Bit	0: Input 1: Output (1)	WO
DP3D	DP_3 Pin I/O Select Bit	0: Input 1: Output (1)	WO
DP4D	DP_4 Pin I/O Select Bit	0: Input 1: Output (1)	WO
DP5D	DP_5 Pin I/O Select Bit	0: Input 1: Output (1)	WO
— (b6)	Reserved	Should be written with 0	WO
— (b7)	Reserved	Should be written with 1	WO

Note:
1. Confirm that the corresponding pin of MCU port P10 is set to input before setting this bit to 1.

Figure 27.15 DPC2 Register

27.4.12 DAP I/O Pin Control Register 3 (DPC3 Register)

DAP I/O Pin Control Register 3

b7	b6	b5	b4	b3	b2	b1	b0
1	0	0	0	0	0		

Symbol	Address	Reset Value
DPC3	13h	0000 0000b

Bit Symbol	Bit Name	Function	RW
DP6D	DP_6 Pin I/O Select Bit	0: Input 1: Output (1)	WO
DP7D	DP_7 Pin I/O Select Bit	0: Input 1: Output (1)	WO
— (b6-b2)	Reserved	Should be written with 0	WO
— (b7)	Reserved	Should be written with 1	WO

Note:
1. Confirm that the corresponding pin of MCU port P10 is set to input before setting this bit to 1.

Figure 27.16 DPC3 Register

27.4.13 DAP I/O Pin Control Register 4 (DPC4 Register)

DAP I/O Pin Control Register 4

b7	b6	b5	b4	b3	b2	b1	b0
1	0	0		0	0	0	0

Symbol	Address	Reset Value
DPC4	14h	0000 0000b

Bit Symbol	Bit Name	Function	RW
— (b3-b0)	Reserved	Should be written with 0	WO
TXDD	DTXD Pin I/O Select Bit (1)	0: Input 1: Output	WO
— (b6-b5)	Reserved	Should be written with 0	WO
— (b7)	Reserved	Should be written with 1	WO

Note:
1. The DTXD is an internal pin to output serial data from DSP to MCU. Set this bit to 1 immediately after an reset is released.

Figure 27.17 DPC4 Register

27.4.14 DAP I/O Pin Control Register 5 (DPC5 Register)

DAP I/O Pin Control Register 5

b7	b6	b5	b4	b3	b2	b1	b0
1	0	0	0	0		0	0

Symbol: DPC5 | Address: 15h | Reset Value: 0000 0000b

Bit Symbol	Bit Name	Function	RW
— (b1-b0)	Reserved	Should be written with 0	WO
SDOS	SDOUT0 to SDOUT2 Output Signal Select Bit	0: Fixed low level output 1: Output of signals SDOUT0 to SDOUT2	WO
— (b6-b3)	Reserved	Should be written with 0	WO
— (b7)	Reserved	Should be written with 1	WO

Figure 27.18 DPC5 Register

SDOS Bit

This bit controls the output signal from pins SDOUT0, SDOUT1, and SDOUT2. When not outputting serial audio data to external devices, the noise can be reduced by setting this bit to 0.

27.4.15 DAP I/O Pin Control Register 6 (DPC6 Register)

DAP I/O Pin Control Register 6

b7	b6	b5	b4	b3	b2	b1	b0
1	0	0	0	0		0	

Symbol	Address	Reset Value
DPC6	16h	0000 0000b

Bit Symbol	Bit Name	Function	RW
CLKS	BITCLK and LRCLK Output Signal Select Bit	0: Fixed low level output 1: Output of signals BITCLK and LRCLK	WO
— (b1)	Reserved	Should be written with 0	WO
TDOS	TDO Output Signal Control Bit	0: Fixed low level output 1: Output of TDO signal	WO
— (b6-b3)	Reserved	Should be written with 0	WO
— (b7)	Reserved	Should be written with 1	WO

Figure 27.19 DPC6 Register

CLKS Bit

This bit controls the output signal from pins BITCLK and LRCLK. When not outputting signals BITCLK and LRC to external devices, the noise can be reduced by setting this bit to 0.

TDOS Bit

This bit controls the output signal from the TDO pin. When not performing a debug, the noise can be reduced by setting this bit to 0.

27.4.16 DAP I/O Pin Control Register 7 (DPC7 Register)

DAP I/O Pin Control Register 7

b7	b6	b5	b4	b3	b2	b1	b0
1	0		0		0		0

Symbol: DPC7 | Address: 17h | Reset Value: 0000 0000b

Bit Symbol	Bit Name	Function	RW
— (b0)	Reserved	Should be written with 0	WO
DP0S	DP_0 Pin Output Signal Select Bit	0: Fixed low level output 1: Output of the DP_0 signal	WO
— (b2)	Reserved	Should be written with 0	WO
DP1S	DP_1 Pin Output Signal Select Bit	0: Fixed low level output 1: Output of the DP_1 signal	WO
— (b4)	Reserved	Should be written with 0	WO
DP2S	DP_2 Pin Output Signal Select Bit	0: Fixed low level output 1: Output of the DP_2 signal	WO
— (b6)	Reserved	Should be written with 0	WO
— (b7)	Reserved	Should be written with 1	WO

Figure 27.20 DPC7 Register

27.4.17 DAP I/O Pin Control Register 8 (DPC8 Register)

DAP I/O Pin Control Register 8

b7	b6	b5	b4	b3	b2	b1	b0
1	0	0	0		0		0

Symbol: DPC8 | Address: 18h | Reset Value: 0000 0000b

Bit Symbol	Bit Name	Function	RW
— (b0)	Reserved	Should be written with 0	WO
DP35S	DP_3, DP_4, and DP_5 Pin Output Signal Select Bit	0: Fixed low level output 1: Output of DP_3, DP_4, and DP_5 signals	WO
— (b2)	Reserved	Should be written with 0	WO
DP67S	DP_6 and DP_7 Pin Output Signal Select Bit	0: Fixed low level output 1: Output of DP_6 and DP_7 signals	WO
— (b6-b4)	Reserved	Should be written with 0	WO
— (b7)	Reserved	Should be written with 1	WO

Figure 27.21 DPC8 Register

27.4.18 DAP I/O Pin Control Register 10 (DPC10 Register)

DAP I/O Pin Control Register 10

b7	b6	b5	b4	b3	b2	b1	b0
1	0	0	0	0	0	0	

Symbol	Address	Reset Value
DPC10	1Ah	0000 0000b

Bit Symbol	Bit Name	Function	RW
TXDS	DTXD Output Signal Select Bit (1)	0: Fixed low level output 1: Output of DTXD signal	WO
— (b6-b2)	Reserved	Should be written with 0	WO
— (b7)	Reserved	Should be written with 1	WO

Note:
1. The DTXD is an internal pin to output serial data from DSP to MCU.

Figure 27.22 DPC10 Register

27.4.19 DAP Input Signal Control Register 0 (DIC0 Register)

DAP Input Signal Control Register 0

b7	b6	b5	b4	b3	b2	b1	b0
1	0		0		0	0	0

Symbol	Address	Reset Value
DIC0	1Ch	0000 0000b

Bit Symbol	Bit Name	Function	RW
— (b2-b0)	Reserved	Should be written with 0	WO
JTAG	JTAG Debug Signal Input Enable Bit	0: Disable JTAG debug signal input 1: Enable JTAG debug signal input	WO
— (b4)	Reserved	Should be written with 0	WO
DPIE	DP_0 to DP_7 Signal Input Enable Bit	0: Disable DP_0 to DP_7 signal input 1: Enable DP_0 to DP_7 signal input	WO
— (b6)	Reserved	Should be written with 0	WO
— (b7)	Reserved	Should be written with 1	WO

Figure 27.23 DIC0 Register

JTAG Bit

This bit enables signal input from the following pins: TDI, TRST, TMS, and TCK. When performing a debug, set this bit to 1.

DPIE Bit

This bit enables input signals from general purpose I/O ports DP_0 to DP_7 to be read from DSP.

27.4.20 DAP Input Signal Control Register 1 (DIC1 Register)

DAP Input Signal Control Register 1

b7	b6	b5	b4	b3	b2	b1	b0
1	0	0	0	0	0	0	0

Symbol: DIC1 | Address: 1Dh | Reset Value: 0000 0000b

Bit Symbol	Bit Name	Function	RW
SDI1E	SDIN1 Signal Input Enable Bit	0: Disable SDIN1 signal input 1: Enable SDIN1 signal input	WO
— (b6-b1)	Reserved	Should be written with 0	WO
— (b7)	Reserved	Should be written with 1	WO

Figure 27.24 DIC1 Register

SDI1E Bit

This bit enables signal input from the SDIN1 pin. When inputting serial audio data from external devices, set this bit to 1.

27.4.21 DAP Protect Register (DPRR Register)

DAP Protect Register

b7	b6	b5	b4	b3	b2	b1	b0
1	0	0	0	0	0	0	

Symbol: DPRR | Address: 1Fh | Reset Value: 0000 0000b

Bit Symbol	Bit Name	Function	RW
WE	Write Enable Bit	0: Write disabled to registers DPLC and DVRC 1: Write enabled to registers DPLC and DVRC	WO
— (b6-b1)	Reserved	Should be written with 0	WO
— (b7)	Reserved	Should be written with 1	WO

Figure 27.25 DPRR Register

WE Bit

This bit is used to protect registers DPLC and DVRC.

When the WE bit is set to 1, these registers become writable. When this bit is set to 0, these registers are protected from being rewritten.

27.4.22 DAP Clock Control Register (DCKC Register)

DAP Clock Control Register

b7	b6	b5	b4	b3	b2	b1	b0
1	1		1	1	1		0

Symbol: DCKC
Address: 2Dh
Reset Value: 0000 0000b

Bit Symbol	Bit Name	Function	RW
— (b0)	Reserved	Should be written with 0	WO
MCSS	MCLK Clock Source Select Bit	0: Input from MCLK pin 1: PLL output divided by 4	WO
— (b4-b2)	Reserved	Should be written with 0	WO
— (b7)	Reserved	Should be written with 1	WO

Figure 27.26 DCKC Register

MCSS Bit

When this bit is 0, a clock provided from the MCLK pin is used as a master clock for the A/D converter and the D/A converter. When this bit is 1, the PLL clock divided by 4 is used. The performance of the A/D converter or D/A converter is better when this bit is set to 0.

27.5 Operation Sequence

This section describes the basic operation sequence of DAP. Figure 27.27 shows a block diagram of power system and Figure 27.28 shows a block diagram of reset and clock system.

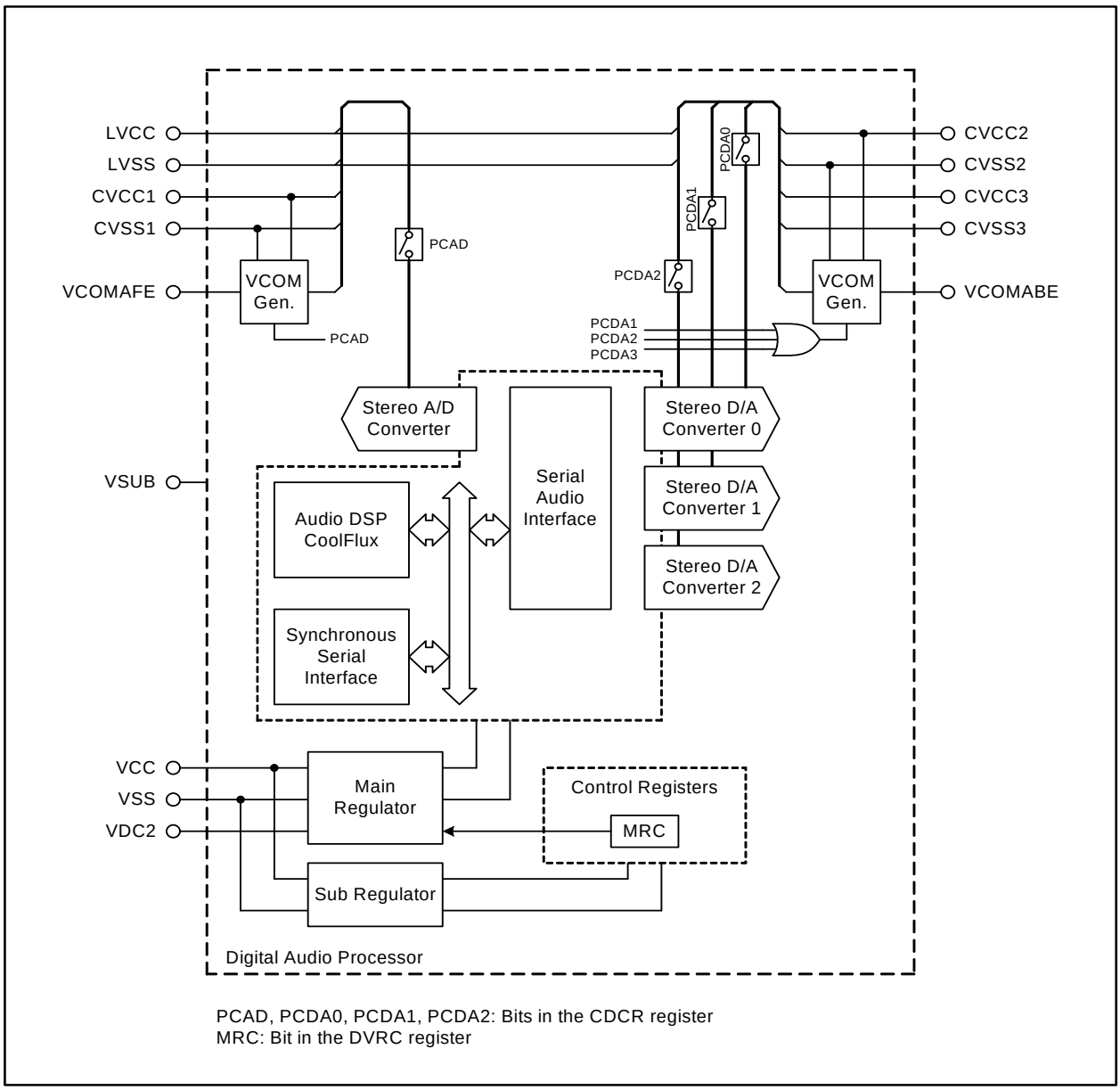


Figure 27.27 DAP Power Supply System

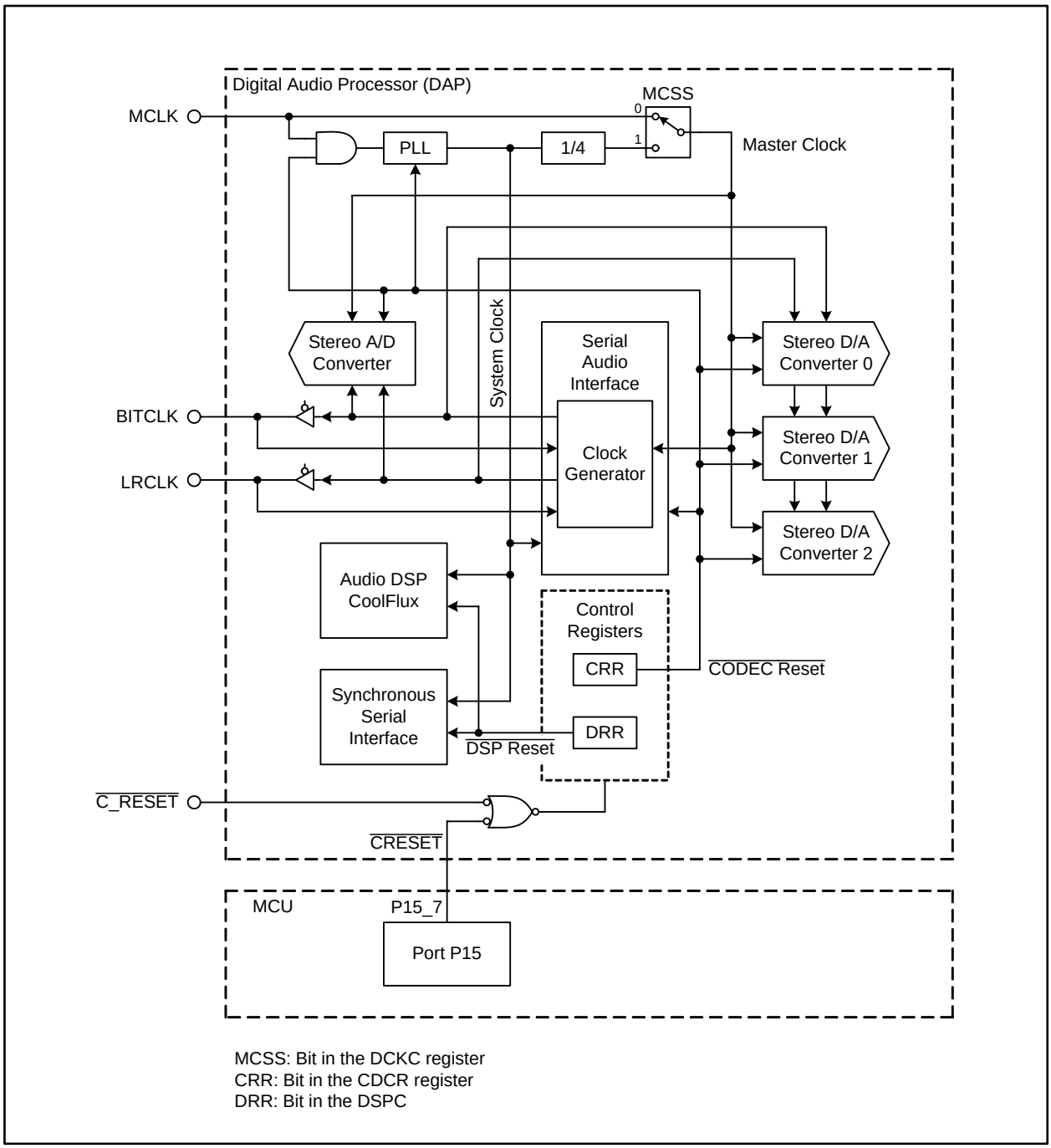


Figure 27.28 Reset and Clock System

27.5.1 Start-up Sequence

When turning on the power for the R32C/102 Group, the main regulator of DAP and any other blocks except control registers are not powered. The power supply of the main regulator and CODEC is required to be controlled by serial communication from the MCU to ports P15 _4 to P15_6.
The following procedure should be done when the power is turned on (refer to the start-up sequence in Figure 27.29):

(1) The power-on-reset is done when the power is turned on.
(2) Set the MRC bit in the DVRC register to 1 (main regulator active).
(3) Hold the state for a period of main regulator output stabilization time (t_{enPU}).
(4) Hold the state for a period of oscillator start-up time (t_{OSC}) for MCLK. Apply the value for t_{OSC} according to the specification of your crystal oscillator.
(5) Set the CRR bit in the CDCR register to 1 (reset released) to release the reset state of CODEC and PLL.
(6) Configure the PLL and set the OSC bit in the DPLC register to 1 (start PLL oscillator running).
(7) Hold the state for a period of PLL oscillator stabilization time ($t_{LOCK(DPLL)}$).
(8) Set the MCSS bit in the DCKC register to 1 (PLL output divided by 4).
(9) Set bits PCAD and PCDA0 to PCDA2 in the CDCR register to 1 (power-on) to supply the power to the A/D converter and D/A converters.
(10) Configure the audio interface, CODEC, and I/O ports.
(11) Set the DRR bit in the DSPC register to 1 (reset released) to release the reset state of DSP.

When the procedure above is completed, the DSP is ready to start an operation in program transfer mode. Transfer the program for DSP from the MCU.
Note that some MCU's programs may cause to generate unexpected waveform at ports P11 and P15 during initial setting. To avoid the operation being affected by this phenomenon, execute the initial setting of MCU while the port P15_7 ($\overline{\text{CRESET}}$) is held low. When the initial setting is completed, drive the port P15_7 high to release the reset state of DAP modules, then execute steps above from (2).

After transferring the DSP programs, execute the following steps:

(12) Set the DRR bit in the DSPC register to 0 (reset) to reset the DSP.
(13) Set the DMOD bit in the DSPC register to 1 (program execute mode) and the DRR bit to 1 (reset released).

Please contact a Renesas sales office regarding the DSP and communication protocol.

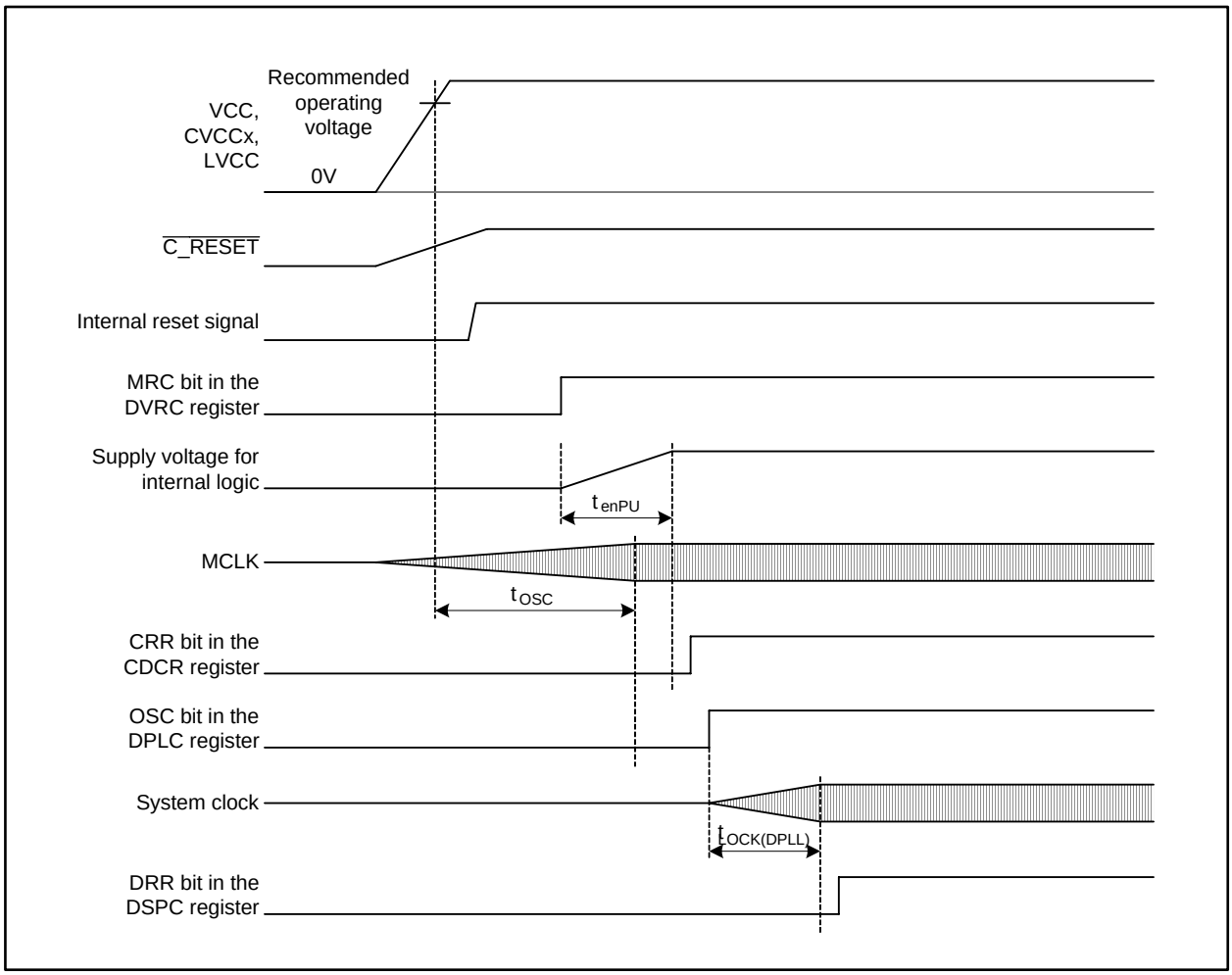


Figure 27.29 Start-up Sequence when Turing On the Power

27.5.2 System Clock Configuration

The system clock, which is the operating clock for DSP, is generated by multiplying MCLK with the PLL frequency synthesizer. As listed in Table 27.5, six types of combination as a PLL configuration are prospective, however some combinations are not available depending on MCLK frequency or the MCSS bit setting in the DCKD register.

Table 27.6 lists the configurable combinations. In the column of “MCSS”, “OK” means an available combination and “NG” is an unavailable one. Consequently, select a combination of MCLK and PLL configuration with which required MCSS bit setting shows “OK”.

Table 27.6 PLL Configuration and System Clock Frequency

fs	MCLK		Reference Clock		PLL	System Clock		Master Clock		MCSS	
kHz	fs	MHz	divided by	MHz	multi-plied by	fs	MHz	fs	MHz	0	1
44.1	256	11.2896	4	2.8224	16	1024	45.1584	256	11.2896	OK	OK
					24	1536	67.7376	384	16.9344	OK	NG
			6	1.8816	—	—	—	—	—	NG	NG
					—	—	—	—	—	NG	NG
	384	16.9344	4	4.2336	16	1536	67.7376	384	16.9344	OK	OK
					24	2304	101.6064	—	—	NG	NG
			6	2.8224	16	1024	45.1584	256	11.2896	OK	NG
					24	1536	67.7376	384	16.9344	OK	OK
48.0	256	12.288	4	3.072	16	1024	49.152	256	12.288	OK	OK
					24	1536	73.728	384	18.432	OK	NG
			6	2.048	16	682.7	—	—	—	NG	NG
					24	1024	49.152	256	12.288	OK	OK
	384	18.432	4	4.068	16	1536	73.728	384	18.432	OK	OK
					24	2304	97.632	—	—	NG	NG
			6	3.072	16	1024	49.152	256	12.288	OK	NG
					24	1536	73.728	384	18.432	OK	OK

27.5.3 Power-off Sequence

To turn off the power of DAP, follow the sequence below:

(1) Set the DRR bit in the DSPC register to 0 (reset) to reset the DSP.
(2) Set bits PCAD and PCDA0 to PCDA2 in the CDCR register to 0 (power-off) to shutdown the A/D converter and D/A converters.
(3) Stop supplying the power to CVCC1 to CVCC3, if necessary.
(4) Set the OSC bit in the DPLC register to 0 (stop PLL oscillator running).
(5) Set the CRR bit in the CDCR register to 0 (reset) to reset the CODEC and PLL.
(6) Set the MRC bit in the DVRC register to 0 (main regulator stopped).

Note that since the CODEC of DAP has no soft mute functions, a click-and-pop noise may occur in the step (2) above. To solve this problem, connect an external mute circuit.

27.5.4 Power Saving

DAP's modules are independently powered and controlled for a reset. Therefore the overall power consumption can be saved by not supplying the power to unused modules.

(1) When not using the A/D converter

Set the PCAD bit in the CDCR register to 0 (power-off), and the A/D converter is shut down.

(2) When not using the D/A converters

Set any or all of the PCDA0 to PCDA2 bits in the CDCR register to 0 (power-off), and the D/A converter is shut down.

(3) When not using the DAP

Follow the procedure shown in 27.5.3 "Power-off Sequence", and the power consumption for DAP is significantly reduced.

When the CRR bit in the CDCR register is set to 0 (reset), the pin assigned as output is held low.

28. Electrical Characteristics

Table 28.1 Absolute Maximum Ratings (1)

Symbol	Characteristic		Condition	Value	Unit
V_{CC}	Supply voltage		$V_{CC} = AV_{CC}$	-0.3 to 6.0	V
AV_{CC}	Analog supply voltage		$V_{CC} = AV_{CC}$	-0.3 to 6.0	V
LV_{CC}	Supply voltage for level shifter of CODEC		—	-0.3 to 6.0	V
CV_{CC1} to CV_{CC3}	Analog supply voltage for CODEC		—	-0.3 to 6.0	V
ΔV_{SS}	Potential difference between V_{SS}-LV_{SS}-CV_{SS1} and CV_{SS3}		—	±0.3	V
V_I	Input voltage	XIN, $\overline{\text{RESET}}$, CNVSS, NSD, V_{REF}, P0_0 to P0_7, P1_0 to P1_7, P2_0 to P2_7, P3_0 to P3_7, P5_0 to P5_3, P8_4 to P8_7, P9_0 to P9_7, P10_0 to P10_7, P11_3, P11_4, P12_0 to P12_7, P13_0 to P13_7, P14_1, P14_3 to P14_6, P15_0 to P15_3, $\overline{\text{C\_RESET}}$, MCLK, BITCLK, LRCLK, SDIN0, SDIN1, TCK, TMS, TRST, TDI		-0.3 to V_{CC} + 0.3	V
		P4_0 to P4_7, P5_4 to P5_7, P6_0 to P6_7, P7_0 to P7_7, P8_0 to P8_3		-0.3 to 6.0	V
		AINL, AINR		-0.3 to CV_{CC1} + 0.3	V
V_O	Output voltage	XOUT, P0_0 to P0_7, P1_0 to P1_7, P2_0 to P2_7, P3_0 to P3_7, P4_0 to P4_7, P5_0 to P5_7, P6_0 to P6_7, P7_0 to P7_7, P8_0 to P8_4, P8_6, P8_7, P9_0 to P9_7, P10_0 to P10_7, P11_3, P11_4, P12_0 to P12_7, P13_0 to P13_7, P14_3 to P14_6, P15_0 to P15_3, BITCLK, LRCLK, SDOUT0 to SDOUT2, TDO		-0.3 to V_{CC} + 0.3	V
		AOUT0L to AOUT2L, AOUT0R to AOUT2R		-0.3 to CV_{CC3} + 0.3	V
P_d	Power consumption		T_a = 25°C	500	mW
—	Operating temperature range			-40 to 85	°C
T_{stg}	Storage temperature range			-65 to 150	°C

Note:

1. Stresses above those listed under “Absolute Maximum Ratings” may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or any other conditions above those indicated in the operational sections of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

Table 28.2 Operating Conditions (1/6) (1)

Symbol	Characteristic		Value			Unit
			Min.	Typ.	Max.	
V_{CC}	Digital supply voltage		3.0	3.3	3.6	V
AV_{CC}	Analog supply voltage			V_{CC}		V
V_{REF}	Reference voltage		3.0		V_{CC}	V
LV_{CC}	Supply voltage for level shifter of CODEC		4.5	5.0	5.5	V
CV_{CC1} to CV_{CC3}	Analog supply voltage for CODEC		4.5	5.0	5.5	V
V_{SS}	Digital ground voltage			0		V
AV_{SS}	Analog ground voltage			0		V
LV_{SS}	Supply voltage for level shifter of CODEC			0		V
CV_{SS1} to CV_{SS3}	Analog supply voltage for CODEC			0		V
dV_{CC}/dt	V_{CC} ramp up rate (V_{CC} < 2.0 V)		0.05			V/ms
V_{IH}	High level input voltage	XIN, $\overline{\text{RESET}}$, CNVSS, NSD, P0_0 to P0_7, P1_0 to P1_7, P2_0 to P2_7, P3_0 to P3_7, P5_0 to P5_3, P8_4 to P8_7 (2), P9_0 to P9_7, P10_0 to P10_7, P11_3, P11_4, P12_0 to P12_7, P13_0 to P13_7, P14_1, P14_3 to P14_6, P15_0 to P15_3, $\overline{\text{C\_RESET}}$, MCLK, BITCLK, LRCLK, SDIN0, SDIN1, TCK, TMS, TRST, TDI	0.8 × V_{CC}		V_{CC}	V
		P4_0 to P4_7, P5_4 to P5_7, P6_0 to P6_7, P7_0 to P7_7, P8_0 to P8_3	0.8 × V_{CC}		6.0	V
V_{IL}	Low level input voltage	XIN, $\overline{\text{RESET}}$, CNVSS, NSD, P0_0 to P0_7, P1_0 to P1_7, P2_0 to P2_7, P3_0 to P3_7, P4_0 to P4_7, P5_0 to P5_7, P6_0 to P6_7, P7_0 to P7_7, P8_0 to P8_7 (2), P9_0 to P9_7, P10_0 to P10_7, P11_3, P11_4, P12_0 to P12_7, P13_0 to P13_7, P14_1, P14_3 to P14_6, P15_0 to P15_3, $\overline{\text{C\_RESET}}$, MCLK, BITCLK, LRCLK, SDIN0, SDIN1, TCK, TMS, TRST, TDI	0		0.2 × V_{CC}	V
T_{opr}	Operating temperature range	P version	-40		85	°C

Notes:
1. The device is operationally guaranteed under these operating conditions.
2. V_{IH} and V_{IL} for P8_7 are specified for P8_7 as a programmable port. These values are not applicable to P8_7 as XCIN.

Table 28.3 Operating Conditions (2/6)
(V_{CC} = 3.0 to 3.6 V, V_{SS} = 0 V, and T_a = T_{opr}, unless otherwise noted) (1)

Symbol	Characteristic		Value (2)			Unit
			Min.	Typ.	Max.	
C_{VDC}	Decoupling capacitance for voltage regulator	Inter-pin voltage: 1.5 V	2.4		10.0	µF
C_{VDC2}	Decoupling capacitance for voltage regulator		2.2		10.0	µF

Notes:
1. The device is operationally guaranteed under these operating conditions.
2. This value should be satisfied with due consideration of every condition as follows: operating temperature, DC bias, aging, etc.

Table 28.4 Operating Conditions (3/6)
(V_{CC} = 3.0 to 3.6 V, V_{SS} = 0 V, and T_a = T_{opr}, unless otherwise noted) (1)

Symbol	Characteristic		Value			Unit
			Min.	Typ.	Max.	
$I_{OH(peak)}$	High level peak output current (2)	P0_0 to P0_7, P1_0 to P1_7, P2_0 to P2_7, P3_0 to P3_7, P4_0 to P4_7, P5_0 to P5_7, P6_0 to P6_7, P7_0 to P7_7, P8_0 to P8_4, P8_6, P8_7, P9_0 to P9_7, P10_0 to P10_7, P11_3, P11_4, P12_0 to P12_7, P13_0 to P13_7, P14_3 to P14_6, P15_0 to P15_3			-10.0	mA
$I_{OH(avg)}$	High level average output current (3)	P0_0 to P0_7, P1_0 to P1_7, P2_0 to P2_7, P3_0 to P3_7, P4_0 to P4_7, P5_0 to P5_7, P6_0 to P6_7, P7_0 to P7_7, P8_0 to P8_4, P8_6, P8_7, P9_0 to P9_7, P10_0 to P10_7, P11_3, P11_4, P12_0 to P12_7, P13_0 to P13_7, P14_3 to P14_6, P15_0 to P15_3			-5.0	mA
$I_{OL(peak)}$	Low level peak output current (2)	P0_0 to P0_7, P1_0 to P1_7, P2_0 to P2_7, P3_0 to P3_7, P4_0 to P4_7, P5_0 to P5_7, P6_0 to P6_7, P7_0 to P7_7, P8_0 to P8_4, P8_6, P8_7, P9_0 to P9_7, P10_0 to P10_7, P11_3, P11_4, P12_0 to P12_7, P13_0 to P13_7, P14_3 to P14_6, P15_0 to P15_3			10.0	mA
$I_{OL(avg)}$	Low level average output current (3)	P0_0 to P0_7, P1_0 to P1_7, P2_0 to P2_7, P3_0 to P3_7, P4_0 to P4_7, P5_0 to P5_7, P6_0 to P6_7, P7_0 to P7_7, P8_0 to P8_4, P8_6, P8_7, P9_0 to P9_7, P10_0 to P10_7, P11_3, P11_4, P12_0 to P12_7, P13_0 to P13_7, P14_3 to P14_6, P15_0 to P15_3			5.0	mA

Notes:
1. The device is operationally guaranteed under these operating conditions.
2. The following conditions should be satisfied:
   - The sum of $I_{OL(peak)}$ of ports P0, P1, P2, P8_6, P8_7, P9, P10, P11, P14, and P15 is 80 mA or less.
   - The sum of $I_{OL(peak)}$ of ports P3, P4, P5, P6, P7, P8_0 to P8_4, P12, and P13 is 80 mA or less.
   - The sum of $I_{OH(peak)}$ of ports P0, P1, P2, and P11 is -40 mA or less.
   - The sum of $I_{OH(peak)}$ of ports P8_6, P8_7, P9, P10, P14, and P15 is -40 mA or less.
   - The sum of $I_{OH(peak)}$ of ports P3, P4, P5, P12, and P13 is -40 mA or less.
   - The sum of $I_{OH(peak)}$ of ports P6, P7, and P8_0 to P8_4 is -40 mA or less.
3. Average value within 100 ms.

Table 28.5 Operating Conditions (4/6)
(V_{CC} = 3.0 to 3.6 V, V_{SS} = 0 V, and $T_a = T_{opr}$, unless otherwise noted) (1)

Symbol	Characteristic	Value			Unit
		Min.	Typ.	Max.	
$f_{(XIN)}$	Main clock oscillator frequency	4		16	MHz
$f_{(XRef)}$	Reference clock frequency	2		4	MHz
$f_{(PLL)}$	PLL clock oscillator frequency	96		128	MHz
$f_{(Base)}$	Base clock frequency			50	MHz
$t_{c(Base)}$	Base clock cycle time	20			ns
$f_{(CPU)}$	CPU operating frequency			50	MHz
$t_{c(CPU)}$	CPU clock cycle time	20			ns
$f_{(BCLK)}$	Peripheral bus clock operating frequency			25	MHz
$t_{c(BCLK)}$	Peripheral bus clock cycle time	40			ns
$f_{(PER)}$	Peripheral clock source frequency			32	MHz
$f_{(XCIN)}$	Sub clock oscillator frequency		32.768	62.5	kHz

Note:
1. The device is operationally guaranteed under these operating conditions.

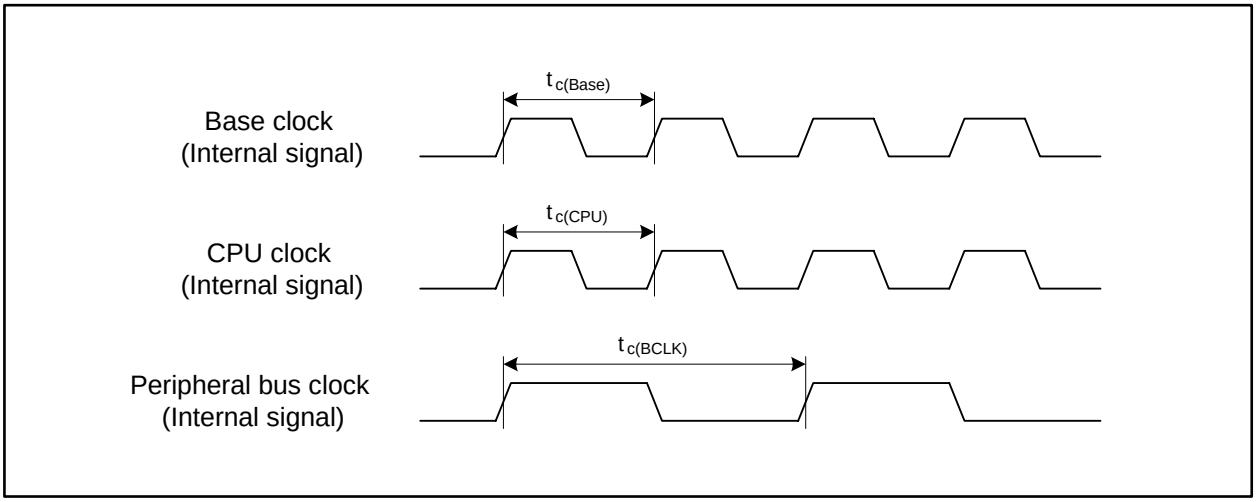


Figure 28.1 Clock Cycle Time

Table 28.6 Operating Conditions (5/6)
(V_{CC} = 3.0 to 3.6 V, V_{SS} = 0 V, and T_a = T_{opr}, unless otherwise noted) (1)

Symbol	Characteristic		Value			Unit
			Min.	Typ.	Max.	
$V_{r(VCC)}$	Allowable ripple voltage	V_{CC} = 3.0 V			0.3	Vp-p
$dV_{r(VCC)}/dt$	Ripple voltage gradient	V_{CC} = 3.0 V			±0.3	V/ms
$f_{r(VCC)}$	Allowable ripple frequency				10	kHz

Note:
1. The device is operationally guaranteed under these operating conditions.

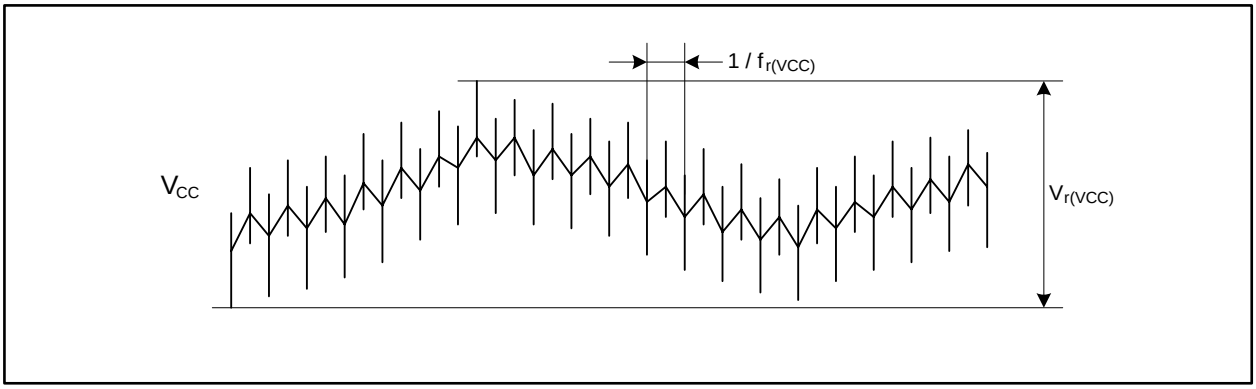


Figure 28.2 Ripple Waveform

Table 28.7 Operating Conditions (6/6)
(V_{CC} = 3.0 to 3.6 V, V_{SS} = 0 V, and T_a = T_{opr}, unless otherwise noted) (1)

Symbol	Characteristic	Value		Unit
		Min.	Max.	
$f_{(MCLK)}$	MCLK input frequency	11.2896	18.432	MHz
$\Delta f / f_{(MCLK)}$	MCLK frequency stability		±100	ppm
$t_{c(MCLK)}$	MCLK input clock cycle time	54.25	88.58	ns
$t_{w(MCLKH)}$	MCLK input high level pulse width	24		ns
$t_{w(MCLKL)}$	MCLK input low level pulse width	24		ns
t_w / t_c	MCLK input duty	40	60	%
$f_{(DREF)}$	DAP PLL reference clock frequency	2	5	MHz
$f_{(DPLL)}$	DAP PLL oscillator frequency	45.1584	73.728	MHz
$f_{(DSP)}$	DSP operating frequency		73.728	MHz

Note:
1. The device is operationally guaranteed under these operating conditions.

Table 28.8 RAM Electrical Characteristics
(V_{CC} = 3.0 to 3.6 V, V_{SS} = 0 V, and Ta = T_{opr}, unless otherwise noted)

Symbol	Characteristic	Measurement condition	Value			Unit
			Min.	Typ.	Max.	
V_{RDR}	RAM data retention voltage	in stop mode	2.0			V

Table 28.9 Flash Memory Electrical Characteristics
(V_{CC} = 3.0 to 3.6 V, V_{SS} = 0 V, and Ta = T_{opr}, unless otherwise noted)

Symbol	Characteristic		Value			Unit
			Min.	Typ.	Max.	
—	Programming and erasure endurance of flash memory (1)	Program area	1000			times
		Data area	10000			times
—	4-word program time	Program area		150	900	µs
		Data area		300	1700	µs
—	Lock bit-program time	Program area		70	500	µs
		Data area		140	1000	µs
—	Block erasure time	4 Kbyte block		0.12	3.0	s
		32 Kbyte block		0.17	3.0	s
		64 Kbyte block		0.20	3.0	s
—	Data retention (2)	T_a = 55°C (3)	10			years

Notes:
1. Program/erase definition
   This value represents the number of erasures per block.
   If the flash memory is programmed/erased n times, each block can be erased n times.
   i.e. If 4-word write is performed in 512 different addresses in the block A of 4 Kbyte and then the block is erased, it is considered the programming/erasure is performed just once.
   However a write in the same address more than once for one erasure is disabled (overwrite disabled).
2. The data retention time includes the periods when the supply voltage is not applied and no clock is provided.
3. Please contact a Renesas Electronics sales office regarding data retention time other than the above.

Table 28.10 Power Supply Circuit Timing Characteristics (V_{CC} = 3.0 to 3.6 V, V_{SS} = 0 V, and $T_a = T_{opr}$, unless otherwise noted)

Symbol	Characteristic	Measurement condition	Value			Unit
			Min.	Typ.	Max.	
$t_{d(P-R)}$	Internal power supply start-up stabilization time after the main power supply is turned on				2	ms

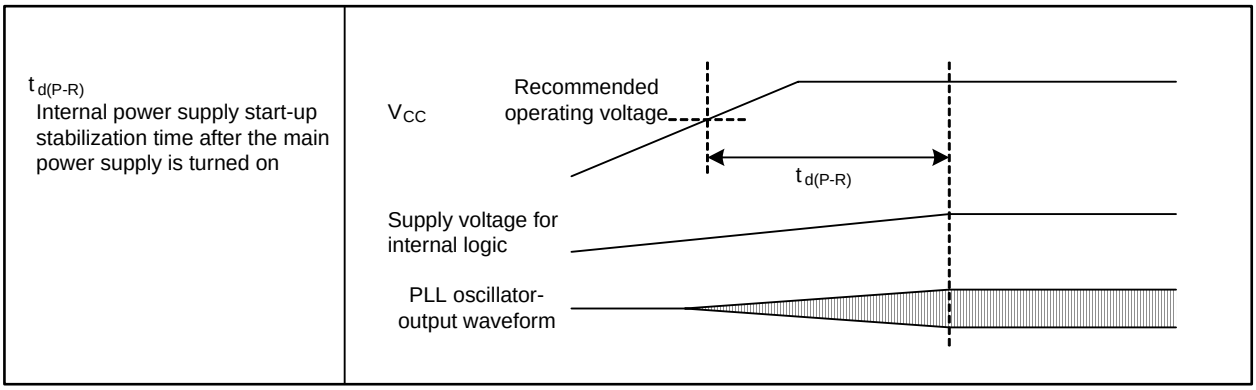

t d(P-R)
Internal power supply start-up stabilization time after the main power supply is turned on
VCC
Recommended operating voltage
t d(P-R)
Supply voltage for internal logic
PLL oscillator-output waveform

Figure 28.3 Power Supply Circuit Timing

Table 28.11 Electrical Characteristics of Voltage Regulator for Internal Logic (V_{CC} = 3.0 to 3.6 V, V_{SS} = 0 V, and $T_a = T_{opr}$, unless otherwise noted)

Symbol	Characteristics	Measurement condition	Value			Unit
			Min.	Typ.	Max.	
V_{VDC1}	Output voltage			1.5		V

Table 28.12 Electrical Characteristics of Oscillator
(V_{CC} = 3.0 to 3.6 V, V_{SS} = 0 V, and T_a = T_{opr}, unless otherwise noted)

Symbol	Characteristics	Measurement condition	Value			Unit
			Min.	Typ.	Max.	
$f_{SO(PLL)}$	PLL clock self-oscillation frequency		35	50	65	MHz
$t_{LOCK(PLL)}$	PLL lock time (1)				1	ms
$t_{jitter(p-p)}$	PLL jitter period (p-p)				2.0	ns
$f_{(OCO)}$	On-chip oscillator frequency		62.5	125	250	kHz

Note:
1. This value is applicable only when the main clock oscillation is stable.

Table 28.13 Electrical Characteristics of Clock Circuitry
(V_{CC} = 3.0 to 3.6 V, V_{SS} = 0 V, and T_a = T_{opr}, unless otherwise noted)

Symbol	Characteristics	Measurement condition	Value			Unit
			Min.	Typ.	Max.	
$t_{rec(WAIT)}$	Recovery time from wait mode to low power mode				225	µs
$t_{rec(STOP)}$	Recovery time from stop mode (1)				225	µs

Note:
1. This recovery time does not include the period until the main clock oscillator is stabilized. The CPU starts operating before the oscillator is stabilized.

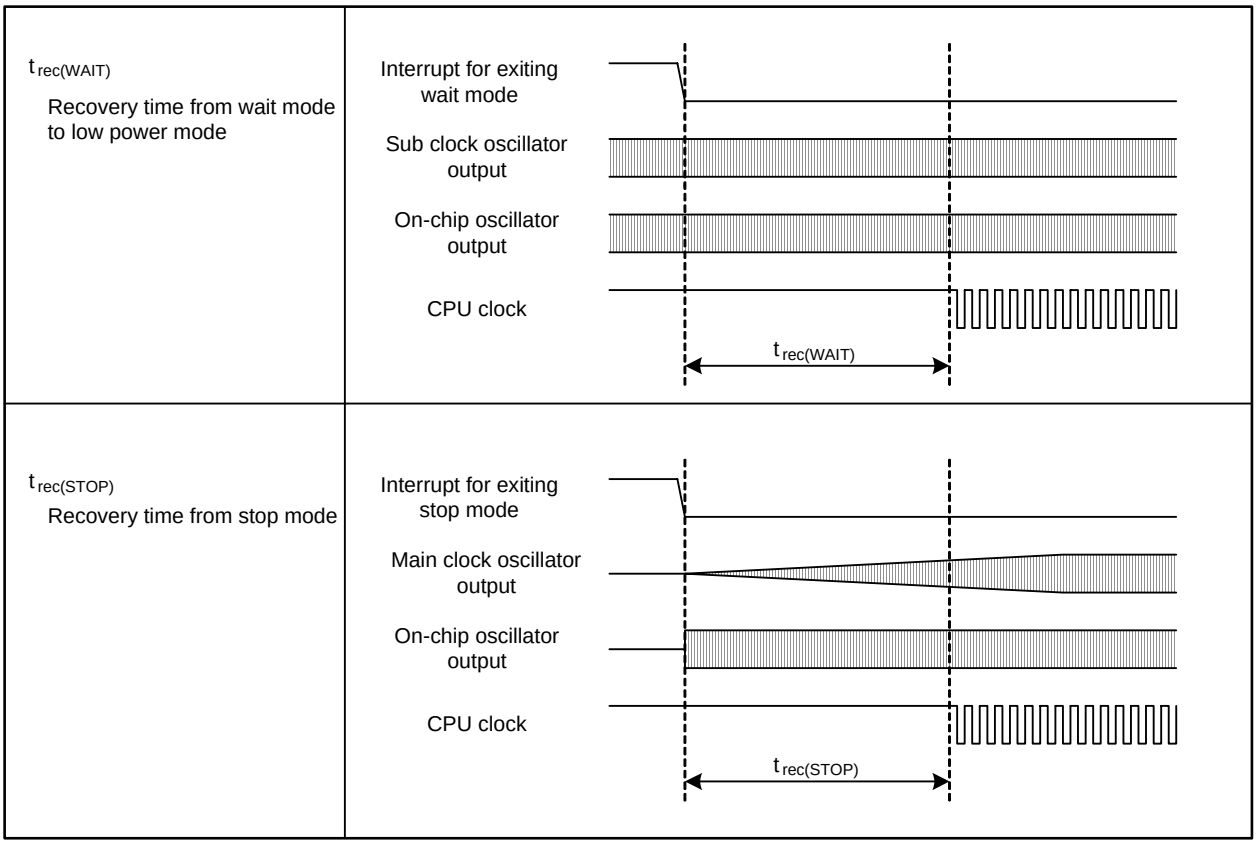


Figure 28.4 Clock Circuit Timing

Table 28.14 Electrical Characteristics of DAP Embedded Regulator (V_{CC} = 3.0 to 3.6 V, V_{SS} = 0 V, and $T_a = T_{opr}$, unless otherwise noted)

Symbol	Characteristics	Measurement condition	Value			Unit
			Min.	Typ.	Max.	
t_{enPU}	Main regulator output stabilization time				2	ms
V_{VDC2}	Main regulator output voltage			1.5		V

Table 28.15 Electrical Characteristics of DAP Embedded PLL Frequency Synthesizer (V_{CC} = 3.0 to 3.6 V, V_{SS} = 0 V, and $T_a = T_{opr}$, unless otherwise noted)

Symbol	Characteristics	Measurement condition	Value			Unit
			Min.	Typ.	Max.	
$t_{LOCK(DPLL)}$	PLL lock time (1)	$f_{(DREF)}$ = 4 MHz			1	ms
$t_{jitterDPLL(PP)}$	PLL jitter period (p-p)				1	ns

Note:
1. This value is applicable only when the MCLK oscillation is stable.

Table 28.16 Electrical Characteristics of Audio CODEC (1): A/D Converter (CV_{CC1} = LV_{CC} = 5.0 V, CV_{SS1} = LV_{SS} = 0 V, T_a = 25°C, fs = 44.1 kHz, f = 1 kHz, unless otherwise noted)

Characteristic	Measurement condition	Value			Unit
		Min.	Typ.	Max.	
Resolution			24		Bits
Full Scale Maximum Input Voltage	THD = 1%	1.0	1.1	1.2	V_{RMS}
Signal-to-noise Ratio	IHF-A, 0 V_{RMS} input, 20 kHz LPF	91	100		dB
Total Harmonic Distortion-plus-Noise to Signal Ratio	IHF-A, 0.8 V_{RMS} input, 20 kHz LPF	80	92		dB
Dynamic Range	IHF-A, -60 dB input, 20 kHz LPF	90	100		dB
Inter-channel Crosstalk	0.8 V_{RMS} input, 1 kHz BPF		-105	-90	dB
Inter-channel Level Difference	0.8 V_{RMS} input			±0.5	dB
Conversion Time (Ts = 1/fs)			28		Ts
Pass-band Ripple	0.8 V_{RMS} input, V_{IN}@1 kHz, from 20 Hz to 20 kHz			±0.5	dB
Input Impedance		23	33		kΩ
Input Bias Voltage			0.5 × CV_{CC1}		V

Table 28.17 Electrical Characteristics of Audio CODEC (2): D/A Converter (CV_{CC2} = CV_{CC3}= LV_{CC} = 5.0 V, CV_{SS2} = CV_{SS3} = LV_{SS} = 0 V, T_a = 25°C, fs = 44.1 kHz, f = 1 kHz, unless otherwise noted)

Characteristic	Measurement condition	Value			Unit
		Min.	Typ.	Max.	
Resolution			24		Bits
Full Scale Maximum Output Voltage	THD = 1%, 0 dBFS input, R_L = 25 kΩ, C_L = 15 pF	1.0	1.1		V_{RMS}
Signal-to-noise Ratio	IHF-A, 20 kHz SPCL	95	100		dB
Total Harmonic Distortion-plus-Noise to Signal Ratio	-2.766 dBFS input, 20 kHz AES17	80	90		dB
Dynamic Range	IHF-A, -60 dB input, 20 kHz SPCL	95	100		dB
Inter-channel Crosstalk	-2.766 dBFS input, 1 kHz BPF		-110	-90	dB
Inter-channel Level Difference	-2.766 dBFS input			±0.5	dB
Conversion Time (Ts = 1/fs)			30		Ts
Pass-band Ripple	-2.766 dBFS input, V_{IN}@1 kHz, from 20 Hz to 20 kHz			±0.5	dB
Output Impedance			110	150	kΩ
Output Bias Voltage			0.5 × CV_{CC3}		V

Timing Requirements (V_{CC} = 3.0 to 3.6 V, V_{SS} = 0 V, and Ta = T_{opr}, unless otherwise noted)

Table 28.18 Flash Memory CPU Rewrite Mode Timing

Symbol	Characteristics	Value		Unit
		Min.	Max.	
t_{cR}	Read cycle time	200		ns
$t_{su(S-R)}$	Chip-select setup time for read	200		ns
$t_{h(R-S)}$	Chip-select hold time after read	0		ns
$t_{su(A-R)}$	Address setup time for read	200		ns
$t_{h(R-A)}$	Address hold time after read	0		ns
$t_{w(R)}$	Read pulse width	100		ns
t_{cW}	Write cycle time	200		ns
$t_{su(S-W)}$	Chip-select setup time for write	0		ns
$t_{h(W-S)}$	Chip-select hold time after write	30		ns
$t_{su(A-W)}$	Address setup time for write	0		ns
$t_{h(W-A)}$	Address hold time after write	30		ns
$t_{w(W)}$	Write pulse width	50		ns

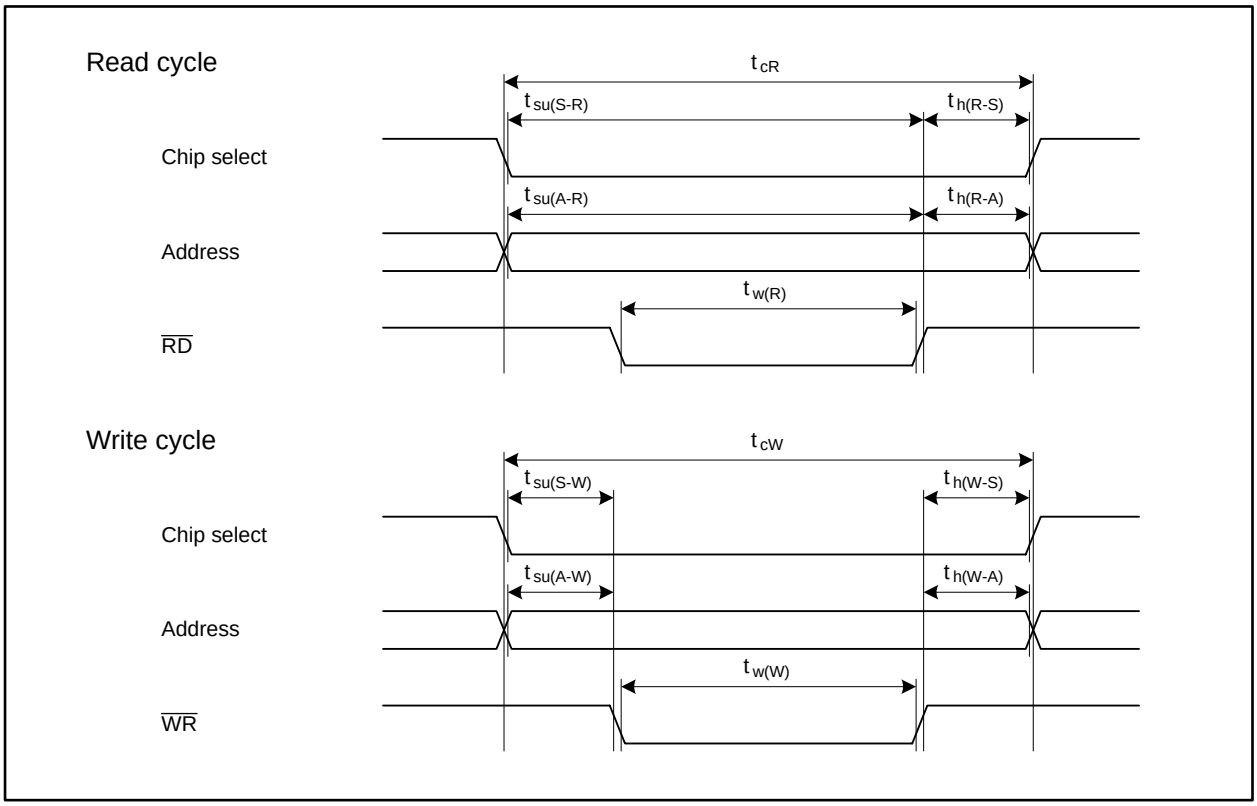


Figure 28.5 Flash Memory CPU Rewrite Mode Timing

V_{CC} = 3.3 V

Table 28.19 Electrical Characteristics (1/4) (V_{CC} = 3.0 to 3.6 V, V_{SS} = 0 V, T_a = T_{opr}, and $f_{(CPU)}$ = 50 MHz, unless otherwise noted)

Symbol	Characteristic		Measurement condition	Value			Unit
				Min.	Typ.	Max.	
V_{OH}	High level output voltage	P0_0 to P0_7, P1_0 to P1_7, P2_0 to P2_7, P3_0 to P3_7, P4_0 to P4_7, P5_0 to P5_7, P6_0 to P6_7, P7_0 to P7_7, P8_0 to P8_4, P8_6, P8_7, P9_0 to P9_7, P10_0 to P10_7, P11_3, P11_4, P12_0 to P12_7, P13_0 to P13_7, P14_3 to P14_6, P15_0 to P15_3, BITCLK, LRCLK, SDOUT0 to SDOUT2, TDO	I_{OH} = -1 mA	V_{CC} - 0.6		V_{CC}	V
V_{OL}	Low level output voltage	P0_0 to P0_7, P1_0 to P1_7, P2_0 to P2_7, P3_0 to P3_7, P4_0 to P4_7, P5_0 to P5_7, P6_0 to P6_7, P7_0 to P7_7, P8_0 to P8_4, P8_6, P8_7, P9_0 to P9_7, P10_0 to P10_7, P11_3, P11_4, P12_0 to P12_7, P13_0 to P13_7, P14_3 to P14_6, P15_0 to P15_3, BITCLK, LRCLK, SDOUT0 to SDOUT2, TDO	I_{OL} = 1 mA			0.5	V

V_{CC} = 3.3 V

Table 28.20 Electrical Characteristics (2/4) (V_{CC} = 3.0 to 3.6 V, V_{SS} = 0 V, T_a = T_{opr}, and $f_{(CPU)}$ = 50 MHz, unless otherwise noted)

Symbol	Characteristic		Measurement condition	Value			Unit
				Min.	Typ.	Max.	
$V_{T+} - V_{T-}$	Hysteresis	$\overline{NMI}$, $\overline{INT0}$ to $\overline{INT8}$, $\overline{KI0}$ to $\overline{KI3}$, TA0IN to TA4IN, TA0OUT to TA4OUT, TB0IN to TB5IN, $\overline{CTS0}$ to $\overline{CTS8}$, CLK0 to CLK8, RXD0 to RXD8, SCL0 to SCL6, SDA0 to SDA6, $\overline{SS0}$ to $\overline{SS6}$, SRXD0 to SRXD6, $\overline{ADTRG}$, IIO0_0 to IIO0_7, IIO1_0 to IIO1_7, UD0A, UD0B, UD1A, UD1B, ISCLK2, ISRXD2, IEIN, CAN0IN, $\overline{CAN0WU}$		0.2		1.0	V
		$\overline{RESET}$		0.2		1.8	V
I_{IH}	High level input current	XIN, $\overline{RESET}$, CNVSS, NSD, P0_0 to P0_7, P1_0 to P1_7, P2_0 to P2_7, P3_0 to P3_7, P4_0 to P4_7, P5_0 to P5_7, P6_0 to P6_7, P7_0 to P7_7, P8_0 to P8_7, P9_0 to P9_7, P10_0 to P10_7, P11_3, P11_4, P12_0 to P12_7, P13_0 to P13_7, P14_1, P14_3 to P14_6, P15_0 to P15_3 , $\overline{C\_RESET}$, MCLK, BITCLK, LRCLK, SDIN0, SDIN1, TCK, TMS, TRST, TDI	V_I = 3.3 V			4.0	µA
I_{IL}	Low level input current	XIN, $\overline{RESET}$, CNVSS, NSD, P0_0 to P0_7, P1_0 to P1_7, P2_0 to P2_7, P3_0 to P3_7, P4_0 to P4_7, P5_0 to P5_7, P6_0 to P6_7, P7_0 to P7_7, P8_0 to P8_7, P9_0 to P9_7, P10_0 to P10_7, P11_3, P11_4, P12_0 to P12_7, P13_0 to P13_7, P14_1, P14_3 to P14_6, P15_0 to P15_3, $\overline{C\_RESET}$, MCLK, BITCLK, LRCLK, SDIN0, SDIN1, TCK, TMS, TRST, TDI	V_I = 0 V			-4.0	µA
R_{PULLUP}	Pull-up resistor	P0_0 to P0_7, P1_0 to P1_7, P2_0 to P2_7, P3_0 to P3_7, P5_0 to P5_3, P8_4, P8_6, P8_7, P9_0 to P9_7, P10_0 to P10_7, P11_3, P11_4, P12_0 to P12_7, P13_0 to P13_7, P14_1, P14_3 to P14_6, P15_0 to P15_3	V_I = 0 V	50	100	500	kΩ
R_{fXIN}	Feedback resistor	XIN			3		MΩ
R_{fXCIN}	Feedback resistor	XCIN			25		MΩ

V_{CC} = 3.3 V

Table 28.21 Electrical Characteristics (3/4)
(V_{CC} = 3.0 to 3.6 V, V_{SS} = 0 V, and T_a = T_{opr}, unless otherwise noted)

Symbol	Characteristic	Measurement condition		Value			Unit
				Min.	Typ.	Max.	
I_{CC}	Power supply current	In single-chip mode, output pins are left open and others are connected to V_{SS} XIN-XOUT Drive power: low XCIN-XCOUT Drive power: low	$f_{(CPU)}$ = 50 MHz, $f_{(BCLK)}$ = 25 MHz, $f_{(XIN)}$ = 8 MHz, Active: XIN, PLL, Stopped: XCIN, OCO $f_{(DSP)}$ = 73.728 MHz		70	120	mA
			$f_{(CPU)}$ = 50 MHz, $f_{(BCLK)}$ = 25 MHz, $f_{(XIN)}$ = 8 MHz, Active: XIN, PLL, Stopped: XCIN, OCO, DAP		32	45	mA
			$f_{(CPU)}$ = $f_{SO(PLL)}$/24 MHz, Active: PLL (self-oscillation), Stopped: XIN, XCIN, OCO, DAP		9		mA
			$f_{(CPU)}$ = $f_{(BCLK)}$ = $f_{(XIN)}$/256 MHz, $f_{(XIN)}$ = 8 MHz, Active: XIN, Stopped: PLL, XCIN, OCO, DAP		670		µA
			$f_{(CPU)}$ = $f_{(BCLK)}$ = 32.768 kHz, Active: XCIN, Stopped: XIN, PLL, OCO, DAP, Main regulator: shutdown		185		µA
			$f_{(CPU)}$ = $f_{(BCLK)}$ = $f_{(OCO)}$/4 kHz, Active: OCO, Stopped: XIN, PLL, XCIN, DAP, Main regulator: shutdown		195		µA
			$f_{(CPU)}$ = $f_{(BCLK)}$ = $f_{(XIN)}$/256 MHz, $f_{(XIN)}$ = 8 MHz, Active: XIN, Stopped: PLL, XCIN, OCO, DAP, T_a = 25°C, Wait mode		500	900	µA
			$f_{(CPU)}$ = $f_{(BCLK)}$ = 32.768 kHz, Active: XCIN, Stopped: XIN, PLL, OCO, DAP, Main regulator: shutdown, T_a = 25°C, Wait mode		13	145	µA
			$f_{(CPU)}$ = $f_{(BCLK)}$ = $f_{(OCO)}$/4 kHz, Active: OCO, Stopped: XIN, PLL, XCIN, DAP, Main regulator: shutdown, T_a = 25°C, Wait mode		15	155	µA
			Stopped: all clocks, DAP Main regulator: shutdown, T_a = 25°C		10	75	µA

V_{CC} = 3.3 V

Table 28.22 Electrical Characteristics (4/4) (V_{CC} = 3.0 to 3.6 V, V_{SS} = 0 V, and T_a = T_{opr}, unless otherwise noted)

Symbol	Characteristic	Measurement condition	Value			Unit
			Min.	Typ.	Max.	
$I_{CC(C)}$	Power supply for CODEC	CVCC1 = CVCC2 = CVCC3 = 5.0 V fs = 44.1 kHz, f = 1 kHz, Active: A/D converter, D/A converter		32		mA
		CVCC1 = CVCC2 = CVCC3 = 5.0 V Power-off: A/D converter, D/A converter, Ta = 25°C		0.1	1	µA
$I_{CC(L)}$	Power supply for level shifter	LVCC = 5.0 V Active: A/D converter, D/A converter		450		µA
		LVCC = 5.0 V Power-off: A/D converter, D/A converter, Ta = 25°C		0.05	1	µA

V_{CC} = 3.3 V

Table 28.23 A/D Conversion Characteristics (V_{CC} = AV_{CC} = V_{REF} = 3.0 to 3.6 V, V_{SS} = AV_{SS} = 0 V, T_a = T_{opr}, and $f_{(BCLK)}$ = 25 MHz, unless otherwise noted)

Symbol	Characteristic	Measurement condition		Value			Unit
				Min.	Typ.	Max.	
—	Resolution	V_{REF} = V_{CC}				10	Bits
—	Absolute error	V_{REF} = V_{CC} = 3.3 V	AN_0 to AN_7, AN0_0 to AN0_7, AN2_0 to AN2_7, AN15_0 to AN15_3, ANEX0, ANEX1			±5	LSB
			External op-amp connection mode			±7	LSB
INL	Integral non-linearity error	V_{REF} = V_{CC} = 3.3 V	AN_0 to AN_7, AN0_0 to AN0_7, AN2_0 to AN2_7, AN15_0 to AN15_3, ANEX0, ANEX1			±5	LSB
			External op-amp connection mode			±7	LSB
DNL	Differential non-linearity error	V_{REF} = V_{CC} = 3.3 V				±1	LSB
—	Offset error					±3	LSB
—	Gain error					±3	LSB
R_{LADDER}	Resistor ladder	V_{REF} = V_{CC}		4		20	kΩ
t_{CONV}	Conversion time (10 bits)	ϕ_{AD} = 10 MHz, with sample and hold function		3.3			µs
t_{CONV}	Conversion time (8 bits)	ϕ_{AD} = 10 MHz, with sample and hold function		2.8			µs
t_{SAMP}	Sampling time	ϕ_{AD} = 10 MHz		0.3			µs
V_{IA}	Analog input voltage			0		V_{REF}	V
ϕ_{AD}	Operating clock frequency	without sample and hold function		0.25		10	MHz
		with sample and hold function		1		10	MHz

V_{CC} = 3.3 V

Table 28.24 D/A Conversion Characteristics (V_{CC} = AV_{CC} = V_{REF} = 3.0 to 3.6 V, V_{SS} = AV_{SS} = 0 V, and T_a = T_{opr}, unless otherwise noted)

Symbol	Characteristic	Measurement condition	Value			Unit
			Min.	Typ.	Max.	
—	Resolution				8	Bits
—	Absolute precision				1.0	%
t_S	Settling time				3	µs
R_O	Output resistance		4	10	20	kΩ
I_{VREF}	Reference input current	(1)			1.0	mA

Note:

1. One D/A converter is used. The DAi register (i = 0, 1) of the other unused converter is set to 00h. The resistor ladder for A/D converter is not considered.
   Even when the VCUT bit in the AD0CON1 register is set to 0 (V_{REF} disconnected), I_{VREF} is supplied.

V_{CC} = 3.3 V

Timing Requirements (V_{CC} = 3.0 to 3.6 V, V_{SS} = 0 V, and T_a = T_{opr}, unless otherwise noted)

Table 28.25 External Clock Input

Symbol	Characteristic	Value		Unit
		Min.	Max.	
$t_{c(X)}$	External clock input period	62.5	250	ns
$t_{w(XH)}$	External clock input high level pulse width	25		ns
$t_{w(XL)}$	External clock input low level pulse width	25		ns
$t_{r(X)}$	External clock input rise time		5	ns
$t_{f(X)}$	External clock input fall time		5	ns
t_w / t_c	External clock input duty	40	60	%

V_{CC} = 3.3 V

Timing Requirements (V_{CC} = 3.0 to 3.6 V, V_{SS} = 0 V, and T_a = T_{opr}, unless otherwise noted)

Table 28.26 Timer A Input (Counting input in event counter mode)

Symbol	Characteristic	Value		Unit
		Min.	Max.	
$t_{c(TA)}$	TAiIN input clock cycle time	200		ns
$t_{w(TAH)}$	TAiIN input high level pulse width	80		ns
$t_{w(TAL)}$	TAiIN input low level pulse width	80		ns

Table 28.27 Timer A Input (Gating input in timer mode)

Symbol	Characteristic	Value		Unit
		Min.	Max.	
$t_{c(TA)}$	TAiIN input clock cycle time	400		ns
$t_{w(TAH)}$	TAiIN input high level pulse width	180		ns
$t_{w(TAL)}$	TAiIN input low level pulse width	180		ns

Table 28.28 Timer A Input (External trigger input in one-shot timer mode)

Symbol	Characteristic	Value		Unit
		Min.	Max.	
$t_{c(TA)}$	TAiIN input clock cycle time	200		ns
$t_{w(TAH)}$	TAiIN input high level pulse width	80		ns
$t_{w(TAL)}$	TAiIN input low level pulse width	80		ns

Table 28.29 Timer A Input (External trigger input in pulse-width modulation mode)

Symbol	Characteristic	Value		Unit
		Min.	Max.	
$t_{w(TAH)}$	TAiIN input high level pulse width	80		ns
$t_{W(TAL)}$	TAiIN input low level pulse width	80		ns

Table 28.30 Timer A Input (Increment/decrement count switching input in event counter mode)

Symbol	Characteristic	Value		Unit
		Min.	Max.	
$t_{c(UP)}$	TAiOUT input clock cycle time	2000		ns
$t_{w(UPH)}$	TAiOUT input high level pulse width	1000		ns
$t_{w(UPL)}$	TAiOUT input low level pulse width	1000		ns
$t_{su(UP-TIN)}$	TAiOUT input setup time	400		ns
$t_{h(TIN-UP)}$	TAiOUT input hold time	400		ns

V_{CC} = 3.3 V

Timing Requirements (V_{CC} = 3.0 to 3.6 V, V_{SS} = 0 V, and T_a = T_{opr}, unless otherwise noted)

Table 28.31 Timer B Input (Counting input in event counter mode)

Symbol	Characteristic	Value		Unit
		Min.	Max.	
$t_{c(TB)}$	TBiIN input clock cycle time (one edge counting)	200		ns
$t_{w(TBH)}$	TBiIN input high level pulse width (one edge counting)	80		ns
$t_{w(TBL)}$	TBiIN input low level pulse width (one edge counting)	80		ns
$t_{c(TB)}$	TBiIN input clock cycle time (both edges counting)	200		ns
$t_{w(TBH)}$	TBiIN input high level pulse width (both edges counting)	80		ns
$t_{w(TBL)}$	TBiIN input low level pulse width (both edges counting)	80		ns

Table 28.32 Timer B Input (Pulse period measure mode)

Symbol	Characteristic	Value		Unit
		Min.	Max.	
$t_{c(TB)}$	TBiIN input clock cycle time	400		ns
$t_{w(TBH)}$	TBiIN input high level pulse width	180		ns
$t_{w(TBL)}$	TBiIN input low level pulse width	180		ns

Table 28.33 Timer B Input (Pulse-width measure mode)

Symbol	Characteristic	Value		Unit
		Min.	Max.	
$t_{c(TB)}$	TBiIN input clock cycle time	400		ns
$t_{w(TBH)}$	TBiIN input high level pulse width	180		ns
$t_{w(TBL)}$	TBiIN input low level pulse width	180		ns

V_{CC} = 3.3 V

Timing Requirements (V_{CC} = 3.0 to 3.6 V, V_{SS} = 0 V, and T_a = T_{opr}, unless otherwise noted)

Table 28.34 Serial Interface

Symbol	Characteristic	Value		Unit
		Min.	Max.	
$t_{c(CK)}$	CLKi input clock cycle time	200		ns
$t_{w(CKH)}$	CLKi input high level pulse width	80		ns
$t_{w(CKL)}$	CLKi input low level pulse width	80		ns
$t_{su(D-C)}$	RXDi input setup time	80		ns
$t_{h(C-D)}$	RXDi input hold time	90		ns

Table 28.35 A/D Trigger Input

Symbol	Characteristic	Value		Unit
		Min.	Max.	
$t_{w(ADH)}$	$\overline{ADTRG}$ input high level pulse width Hardware trigger input high level pulse width	$\frac{3}{\phi_{AD}}$		ns
$t_{w(ADL)}$	$\overline{ADTRG}$ input low level pulse width Hardware trigger input high level pulse width	125		ns

Table 28.36 External Interrupt $\overline{INTi}$ Input

Symbol	Characteristic		Value		Unit
			Min.	Max.	
$t_{w(INH)}$	$\overline{INTi}$ input high level pulse width	Edge sensitive	250		ns
		Level sensitive	$t_{c(CPU)}$ + 200		ns
$t_{W(INL)}$	$\overline{INTi}$ input low level pulse width	Edge sensitive	250		ns
		Level sensitive	$t_{c(CPU)}$ + 200		ns

Table 28.37 Intelligent I/O

Symbol	Characteristic	Value		Unit
		Min.	Max.	
$t_{C(ISCLK2)}$	ISCLK2 input clock cycle time	600		ns
$t_{W(ISCLK2H)}$	ISCLK2 input high level pulse width	270		ns
$t_{W(ISCLK2L)}$	ISCLK2 input low level pulse width	270		ns
$t_{su(RXD-ISCLK2)}$	ISRXD2 input setup time	150		ns
$t_{h(ISCLK2-RXD)}$	ISRXD2 input hold time	100		ns

V_{CC} = 3.3 V

Timing Requirements (V_{CC} = 3.0 to 3.6 V, V_{SS} = 0 V, and T_a = T_{opr}, unless otherwise noted)

Table 28.38 Multi-master I²C-bus Interface

Symbol	Characteristic	Value				Unit
		Standard-mode		Fast-mode		
		Min.	Max.	Min.	Max.	
$t_{w(SCLH)}$	MSCL input high level pulse width	600		600		ns
$t_{w(SCLL)}$	MSCL input low level pulse width	600		600		ns
$t_{r(SCL)}$	MSCL input rise time		1000		300	ns
$t_{f(SCL)}$	MSCL input fall time		300		300	ns
$t_{r(SDA)}$	MSDA input rise time		1000		300	ns
$t_{f(SDA)}$	MSDA input fall time		300		300	ns
$t_{h(SDA-SCL)S}$	MSCL high level hold time after start condition/restart condition	(1)		$2 \times t_{c(\phi IIC)} + 40$		ns
$t_{su(SCL-SDA)P}$	MSCL high level setup time for restart condition/stop condition	(1)		$2 \times t_{c(\phi IIC)} + 40$		ns
$t_{w(SDAH)P}$	MSDA high level pulse width after stop condition	(1)		$4 \times t_{c(\phi IIC)} + 40$		ns
$t_{su(SDA-SCL)}$	MSDA input setup time	100		100		ns
$t_{h(SCL-SDA)}$	MSDA input hold time	0		0		ns

Note:

1. The value is calculated by the following formulas based on a value SSC set by bits SSC4 to SSC0 in the I2CSSCR register:

   $t_{h(SDA-SCL)S} = SSC \div 2 \times t_{c(\phi IIC)} + 40$ [ns]

   $t_{su(SCL-SDA)P} = (SSC \div 2 + 1) \times t_{c(\phi IIC)} + 40$ [ns]

   $t_{w(SDAH)P} = (SSC + 1) \times t_{c(\phi IIC)} + 40$ [ns]

V_{CC} = 3.3 V

Timing Requirements (V_{CC} = 3.0 to 3.6 V, V_{SS} = 0 V, and T_a = T_{opr}, unless otherwise noted)

Table 28.39 Digital Audio Processor (DAP)

Symbol	Characteristic	Value		Unit
		Min.	Max.	
$t_{su(D-BCK)}$	Data setup time for BITCLK (output)	125		ns
$t_{h(BCK-D)}$	Data hold time after BITCLK (output)	0		ns
$t_{w(CRST)}$	Reset pulse width	400		ns
$t_{c(CCLK)}$	CCLK input clock cycle time	200		ns
$t_{su(D-BCK)}$	Data setup time for CCLK	20		ns
$t_{h(CCLK-D)}$	Data hold time after CCLK	0		ns
$t_{su(W-CCLK)}$	$\overline{CWE}$ setup time for CCLK	100		ns
$t_{h(CCLK-W)}$	$\overline{CWE}$ hold time after CCLK	100		ns
$t_{rec(W)}$	$\overline{CWE}$ recovery time	100		ns
$t_{c(DCLK)}$	DLCK input clock cycle time	200		ns
$t_{su(D-DCLK)}$	Data setup time for DCLK	20		ns
$t_{h(DCLK-D)}$	Data hold time after DCLK	0		ns

V_{CC} = 3.3 V

Switching Characteristics (V_{CC} = 3.0 to 3.6 V, V_{SS} = 0 V, and T_a = T_{opr}, unless otherwise noted)

Table 28.40 Serial Interface

Symbol	Characteristic	Measurement condition	Value		Unit
			Min.	Max.	
$t_{d(C-Q)}$	TXDi output delay time	Refer to Figure 28.6		80	ns
$t_{h(C-Q)}$	TXDi output hold time		0		ns

Table 28.41 Intelligent I/O

Symbol	Characteristic	Measurement condition	Value		Unit
			Min.	Max.	
$t_{d(ISCLK2-TXD)}$	ISTXD2 output delay time	Refer to Figure 28.6		180	ns
$t_{h(ISCLK2-RXD)}$	ISTXD2 output hold time		0		ns

Table 28.42 Multi-master I²C-bus Interface (Standard-mode)

Symbol	Characteristic	Measurement condition	Value		Unit
			Min.	Max.	
$t_{f(SCL)}$	MSCL output fall time	Refer to Figure 28.6	2		ns
$t_{f(SDA)}$	MSDA output fall time		2		ns
$t_{d(SDA-SCL)S}$	MSCL output delay time after start condition/restart condition		20 × $t_{c(\phi IIC)}$ - 120	52 × $t_{c(\phi IIC)}$ - 40	ns
$t_{d(SCL-SDA)P}$	Restart condition/stop condition output delay time after MSCL becomes high		20 × $t_{c(\phi IIC)}$ + 40	52 × $t_{c(\phi IIC)}$ + 120	ns
$t_{d(SCL-SDA)}$	MSDA output delay time		2 × $t_{c(\phi IIC)}$ + 40	3 × $t_{c(\phi IIC)}$ + 120	ns

Table 28.43 Multi-master I²C-bus Interface (Fast-mode)

Symbol	Characteristic	Measurement condition	Value		Unit
			Min.	Max.	
$t_{f(SCL)}$	MSCL output fall time	Refer to Figure 28.6	2 (1)		ns
$t_{f(SDA)}$	MSDA output fall time		2 (1)		ns
$t_{d(SDA-SCL)S}$	MSCL output delay time after start condition/restart condition		10 × $t_{c(\phi IIC)}$ - 120	26 × $t_{c(\phi IIC)}$ - 40	ns
$t_{d(SCL-SDA)P}$	Restart condition/stop condition output delay time after MSCL becomes high		10 × $t_{c(\phi IIC)}$ + 40	26 × $t_{c(\phi IIC)}$ + 120	ns
$t_{d(SCL-SDA)}$	MSDA output delay time		2 × $t_{c(\phi IIC)}$ + 40	3 × $t_{c(\phi IIC)}$ + 120	ns

Note:
1. External circuits are required to satisfy the I²C-bus specification.

V_{CC} = 3.3 V

Switching Characteristics (V_{CC} = 3.0 to 3.6 V, V_{SS} = 0 V, and T_a = T_{opr}, unless otherwise noted)

Table 28.44 Digital Audio Processor (DAP)

Symbol	Characteristic	Measurement condition	Value		Unit
			Min.	Max.	
$t_{d(BCK-Q)}$	Data output delay time after BITCLK	Refer to Figure 28.6		10	ns
$t_{h(BCK-Q)}$	Data output hold time after BITCLK		-10		ns
$t_{d(DCK-Q)}$	Data output delay time after DCLK			20	ns
$t_{h(DCK-Q)}$	Data output hold time after DCLK		0		ns

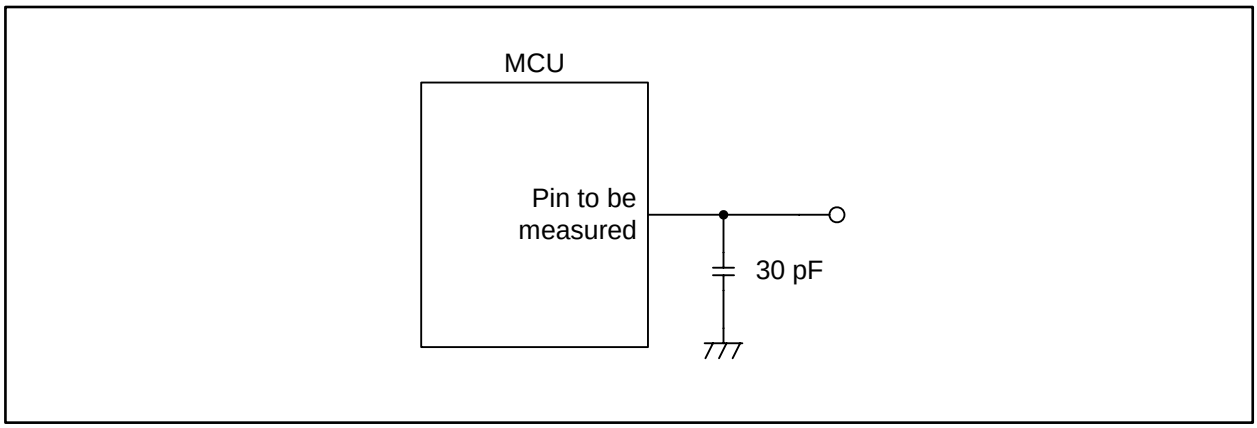


Figure 28.6 Switching Characteristic Measurement Circuit

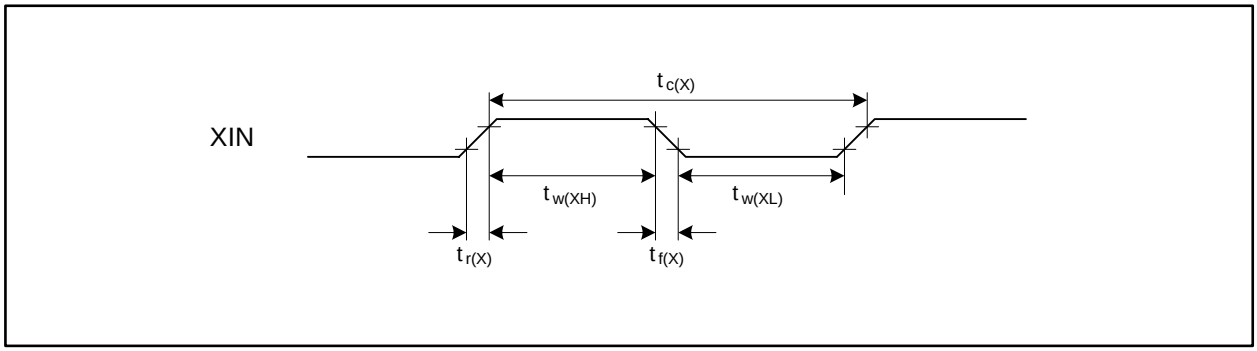


Figure 28.7 External Clock Input Timing

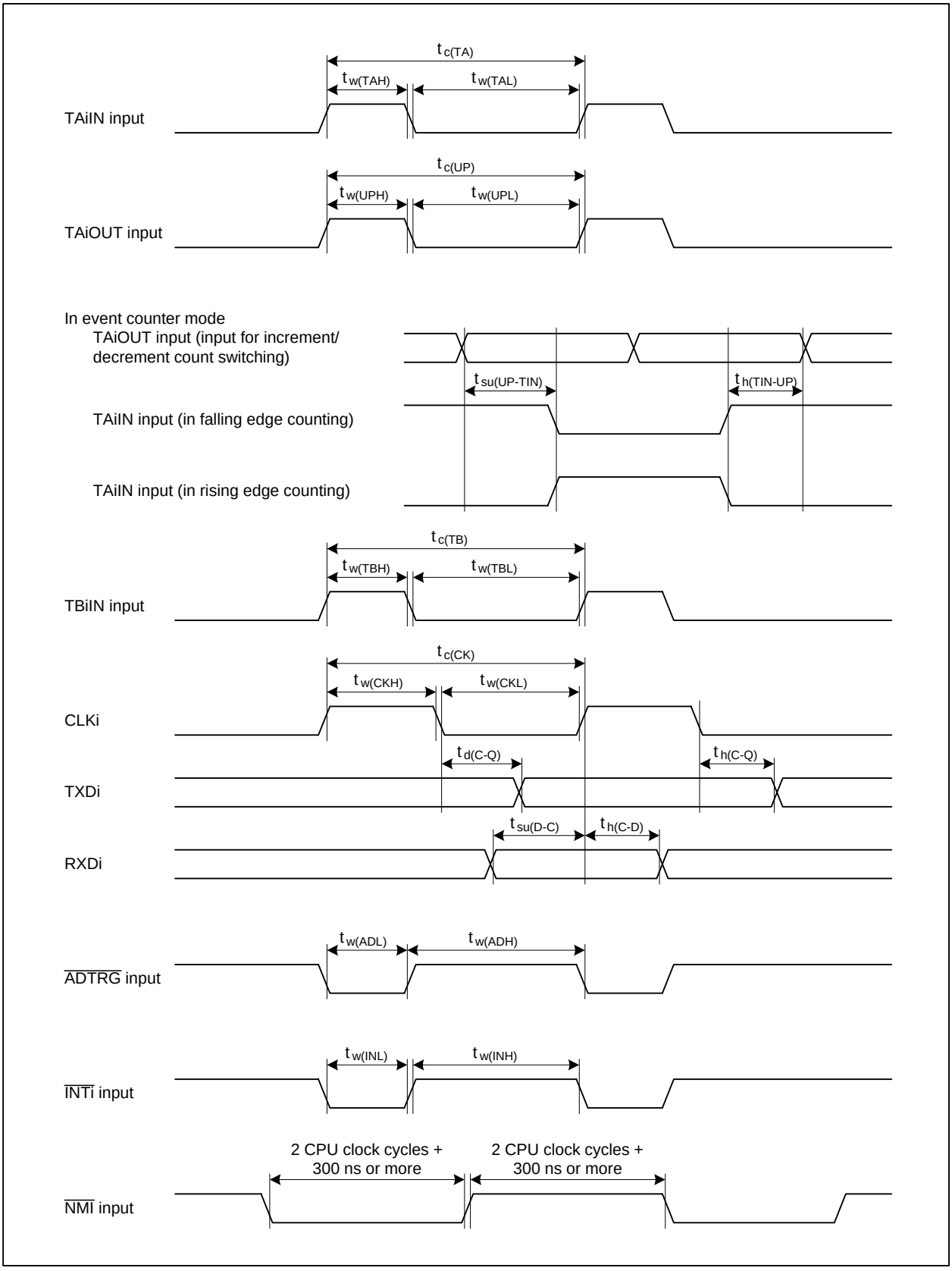


Figure 28.8 Timing of Peripheral Functions

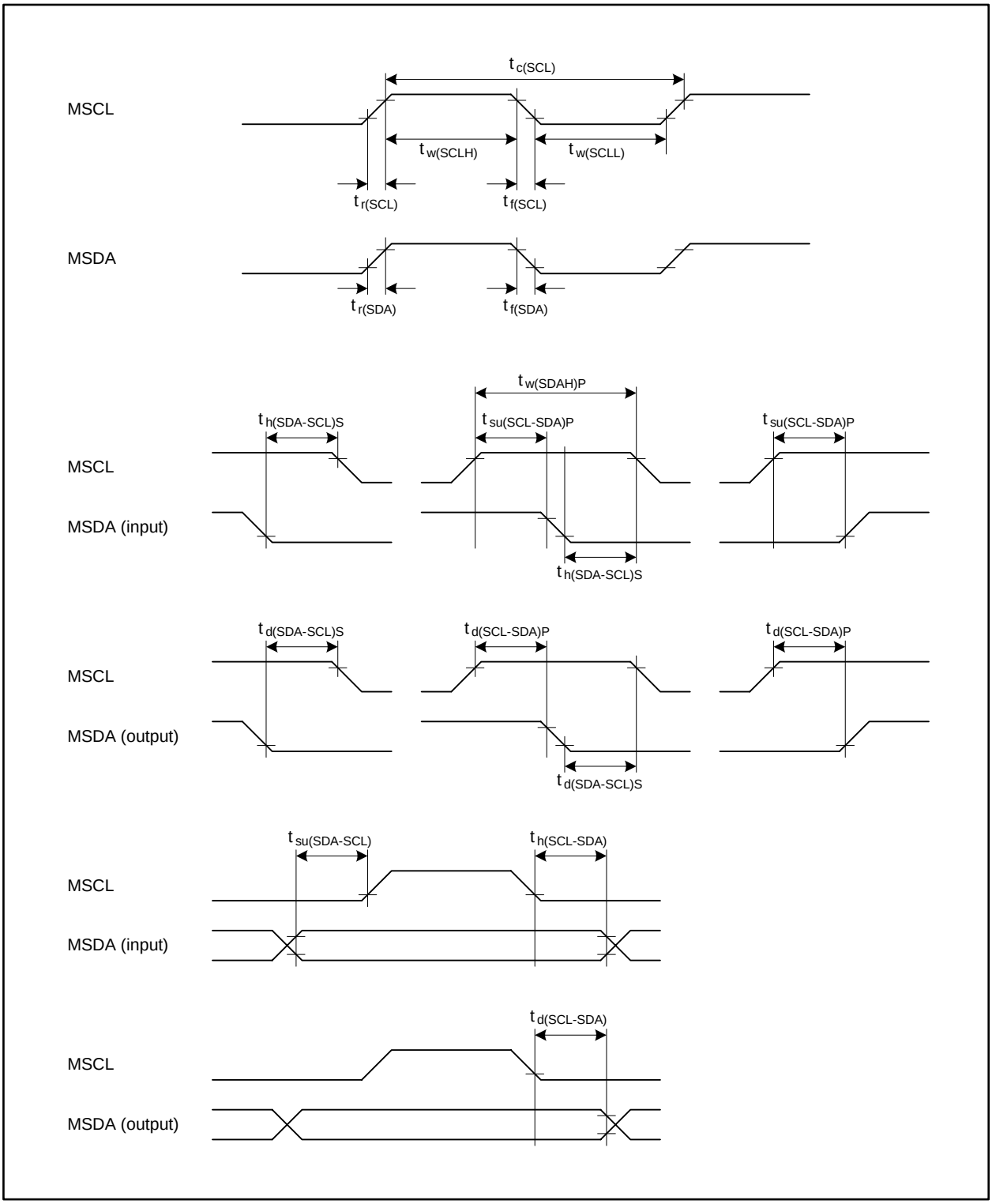


Figure 28.9 Timing of Multi-master I²C-bus Interface

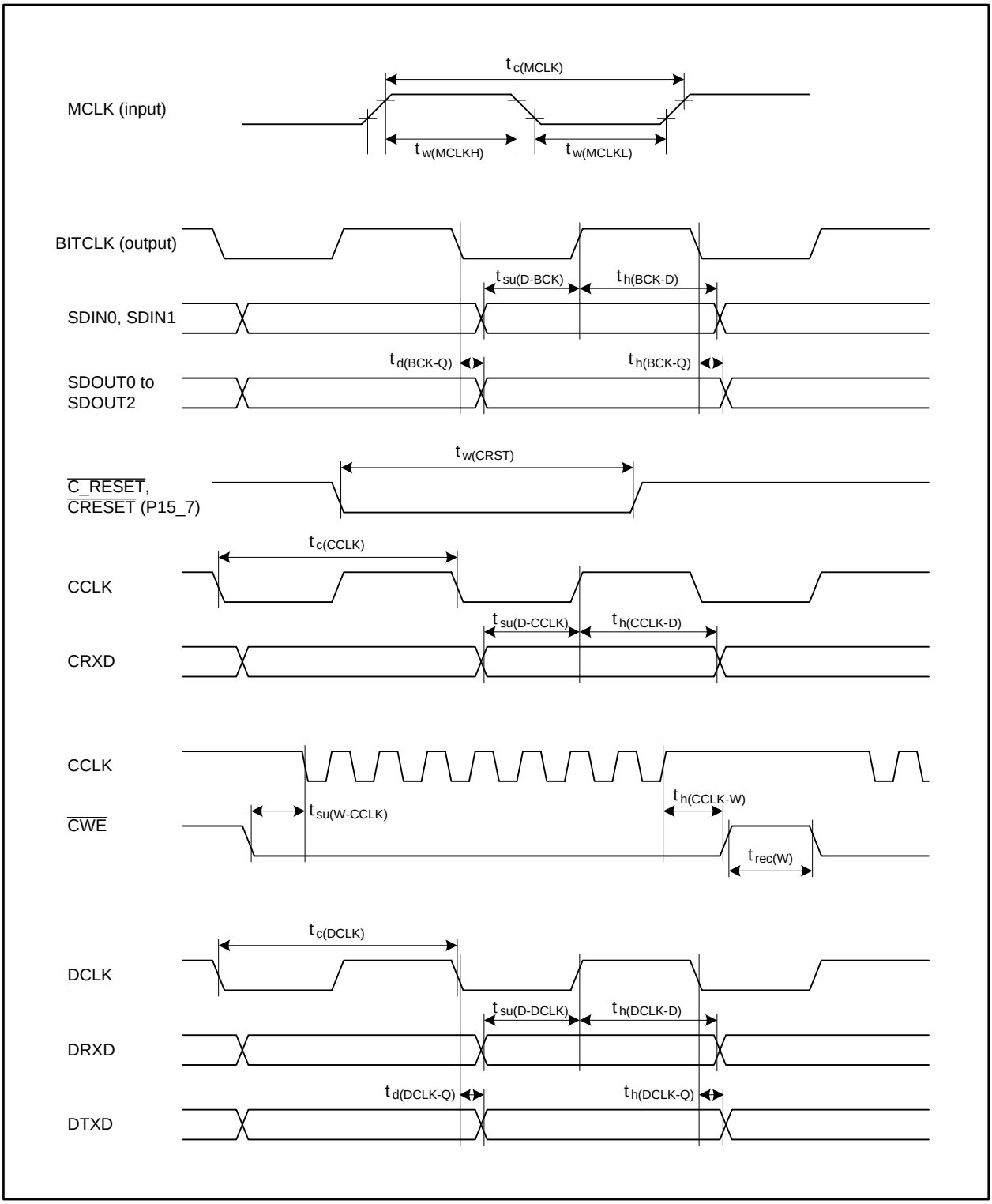


Figure 28.10 Timing of Digital Audio Processor (DAP)

29. Usage Notes

29.1 Notes on Board Designing

29.1.1 Power Supply Pins

The board should be designed so that there is no potential difference between pins with the same name. Note the following points:

- Connect all VSS pins to an identical GND. The traces for the pins should be as wide as physically possible so that the same voltage can be applied to every VSS pin.
- Connect all VCC pins to an identical power supply. The traces for the pins should be as wide as physically possible so that the same voltage can be applied to every VCC pin.

Insert a capacitor between each VCC pin and the VSS pin to ensure the noise tolerance. The capacitor should be beneficially effective at high/low frequencies and should have around 0.1 µF of capacitance. The traces for the capacitor and the power supply pins should be short and wide as much as physically possible.

29.1.2 Supply Voltage

The device is operationally guaranteed under operating conditions specified in electrical characteristics.

Drive the $\overline{\text{RESET}}$ pin low before the supply voltage becomes lower than the recommended value.

29.2 Notes on Register Setting

29.2.1 Registers with Write-only Bits

Table 29.1 lists registers containing write-only bits. For the setting of these registers, read-modify-write instructions listed in Table 29.2 cannot be used since each of these instructions reads the value of an address, modifies the value, and writes to the same address. To set a new value by modifying the previous one, write the previous value into RAM as well as to the register, change the contents of the RAM and then transfer the new value to the register by the MOV instruction.

Table 29.1 Registers with Write-only Bit(s)

Module	Register	Symbol	Address(es)
Watchdog timer	Watchdog timer start register	WDTS	04404Eh
Timer A	Timer A0 register (1)	TA0	0347h-0346h
	Timer A1 register (1)	TA1	0349h-0348h
	Timer A2 register (1)	TA2	034Bh-034Ah
	Timer A3 register (1)	TA3	034Dh-034Ch
	Timer A4 register (1)	TA4	034Fh-034Eh
	Increment/decrement counting select register	UDF	0344h
Three-phase motor control timers	Timer B2 interrupt generating frequency set counter	ICTB2	030Dh
	Timer A1-1 register	TA11	0303h-0302h
	Timer A2-1 register	TA21	0305h-0304h
	Timer A4-1 register	TA41	0307h-0306h
	Dead time timer	DTT	030Ch
Serial interface	UART0 bit rate register	U0BRG	0369h
	UART1 bit rate register	U1BRG	02E9h
	UART2 bit rate register	U2BRG	0339h
	UART3 bit rate register	U3BRG	0329h
	UART4 bit rate register	U4BRG	02F9h
	UART5 bit rate register	U5BRG	01C9h
	UART6 bit rate register	U6BRG	01D9h
	UART7 bit rate register	U7BRG	01E1h
	UART8 bit rate register	U8BRG	01E9h
	UART0 transmit buffer register	U0TB	036Bh-036Ah
	UART1 transmit buffer register	U1TB	02EBh-02EAh
	UART2 transmit buffer register	U2TB	033Bh-033Ah
	UART3 transmit buffer register	U3TB	032Bh-032Ah
	UART4 transmit buffer register	U4TB	02FBh-02FAh
	UART5 transmit buffer register	U5TB	01CBh-01CAh
	UART6 transmit buffer register	U6TB	01DBh-01DAh
	UART7 transmit buffer register	U7TB	01E3h-01E2h
	UART8 transmit buffer register	U8TB	01EBh-01EAh
Intelligent I/O	Group 2 SIO transmit buffer register	G2TB	016Dh-016Ch
CAN module	CAN0 receive FIFO pointer control register	C0RFPCR	047F49h
	CAN0 transmit FIFO pointer control register	C0TFPCR	047F4Bh

Note:
1. The register has write-only bits in one-shot timer mode and pulse-width modulation mode.

Table 29.2 Read-modify-write Instructions

Function	Mnemonic
Transfer	MOV*Dir*
Bit processing	BCLR, BM*Cnd*, BNOT, BSET, BTSTC, and BTSTS
Shifting	ROLC, RORC, ROT, SHA, and SHL
Arithmetic operation	ABS, ADC, ADCF, ADD, ADSF, DEC, DIV, DIVU, DIVX, EXTS, EXTZ, INC, MUL, MULU, NEG, SBB, and SUB
Decimal operation	DADC, DADD, DSBB, and DSUB
Floating-point operation	ADDF, DIVF, MULF, and SUBF
Logical operation	AND, NOT, OR, and XOR

29.3 Notes on Clock Generator

29.3.1 Sub Clock

29.3.1.1 Oscillator Constant Matching

The constant matching of the sub clock oscillator should be evaluated in both cases when the drive power is high and low.
Contact the oscillator manufacturer for details on the oscillation circuit constant matching.

29.3.2 Power Control

Do not switch the base clock source until the oscillation of the clock to be used has stabilized. However, this does not apply to the on-chip oscillator since it starts running immediately after the CM31 bit in the CM3 register is set to 1.
To switch the base clock source from the PLL clock to a low speed clock, use the MOV.L or OR.L instruction to set the BCS bit in the CCR register to 1.

- Program example in assembly language

```
OR.L    #80h, 0004h
```

- Program example in C language

```
asm("OR.L  #80h, 0004h");
```

29.3.2.1 Stop Mode

- To exit stop mode using a reset, apply a low signal to the $\overline{\text{RESET}}$ pin until the main clock oscillation stabilizes.

29.3.2.2 Suggestions for Power Saving

The followings are suggestions to reduce power consumption when programming or designing systems.

- I/O pins:
  If inputs are floating, both transistors may be conducting. Set unassigned pins to input mode and connect each of them to VSS via a resistor, or set them to output mode and leave them open.
- A/D converter:
  When not performing the A/D conversion, set the VCUT bit in the AD0CON1 register to 0 (VREF disconnected). To perform the A/D conversion, set the VCUT bit to 1 (VREF connected) and wait 1 µs or more before starting conversion.
- D/A converter:
  When not performing the D/A conversion, set the DAiE bit in the DACON register (i = 0, 1) to 0 (output disabled) and the DAi register to 00h.
- Peripheral clock stop:
  When entering wait mode, power consumption can be reduced by setting the CM02 bit in the CM0 register to 1 to stop the peripheral clock source. However, this setting does not stop the fC32.

29.4 Notes on Interrupts

29.4.1 ISP Setting

The interrupt stack pointer (ISP) is initialized to 00000000h after a reset. Set a value to the ISP before an interrupt is accepted, otherwise the program may go out of control. A multiple of 4 should be set to the ISP, which enables faster interrupt sequence due to less memory access.
For the use of NMI, in particular, since this interrupt cannot be disabled, the PM24 bit in the PM2 register should be set to 1 (NMI enabled) after the ISP is set at the beginning of program.

29.4.2 NMI

- The NMI cannot be disabled once the PM24 bit in the PM2 register is set to 1 (NMI enabled). This bit setting should be done only for the use of NMI.

- When the PM24 bit in the PM2 register is set to 1 (NMI enabled), the P8_5 bit in the P8 register is enabled just for monitoring the $\overline{\text{NMI}}$ pin state. It is not enabled as a general port.

29.4.3 External Interrupt

- The input signal to the $\overline{\text{INTi}}$ pin (i = 0 to 8) requires the pulse width specified by the electrical characteristics. If a pulse width is narrower than the specification, the external interrupt may not be accepted.

- When the effective level and/or edge of $\overline{\text{INTi}}$ pin (i = 0 to 8) are/is changed by the following bits: bits POL and/or LVS in the INTiIC register, the IFSR0i bit (i = 0 to 5) in the IFSR0 register, and/or the IFSR1j bit (j = i - 6; i = 6 to 8) in the IFSR1 register, the corresponding IR bit may become 1 (interrupt requested). When setting the above mentioned bits, preset bits ILVL2 to ILVL0 in the INTiIC register to 000b (interrupt disabled). After setting the above mentioned bits, set the corresponding IR bit to 0 (no interrupt requested), then set bits ILVL2 to ILVL0.

- The interrupt input signals to pins $\overline{\text{INT6}}$ to $\overline{\text{INT8}}$ are also connected to bits INT6R to INT8R in registers IIO9IR to IIO11IR. Therefore, these input signals, when assigned to the intelligent I/O, can be used as a source for exiting wait mode or stop mode. Note that these signals are enabled only on the falling edge and not affected by the following bit settings: bits POL and LVS in the INTiIC register (i = 0 to 8), IFSR0i bit (i = 0 to 5) in the IFSR0 register, and the IFSR1j bit (j = i - 6; i = 6 to 8) in the IFSR1 register.

29.5 Notes on DMAC

29.5.1 DMAC-associated Register Settings

- Set the DMAC-associated registers while bits MDi1 and MDi0 (i = 0 to 3) in the DMDi register are 00b (DMA transfer disabled). Then, set bits MDi1 and MDi0 to 01b (single transfer) or 11b (repeat transfer) at the end of the setup procedure. This procedure is also applied to rewriting bits UDAi, USAi, and BWi1 and BWi0 in the DMDi register.

- In case the DMAC-associated registers are to be rewritten while DMA transfer is enabled, disable the peripheral function as DMA request source so that no DMA transfer request is generated, then set bits MDi1 and MDi0 in the DMDi register of the corresponding channel to 00b (DMA transfer disabled).

- Once a DMA transfer request is accepted, DMA transfer cannot be disabled even if setting bits MDi1 and MDi0 in the DMDi register to 00b (DMA transfer disabled). Do not change the settings of any DMAC-associated registers other than bits MDi1 and MDi0 until the DMA transfer is completed.

- Wait six or more peripheral bus clocks to set bits MDi1 and MDi0 in the DMDi register to 01b (single transfer) or 11b (repeat transfer) after setting registers DMiSL and DMiSL2.

29.5.2 Read from DMAC-associated Registers

- To sequentially read respective registers DMiSL and DMiSL2, follow the reading order as below:
  DM0SL, DM1SL, DM2SL, and DM3SL
  DM0SL2, DM1SL2, DM2SL2, and DM3SL2

29.6 Notes on Timers

29.6.1 Timer A and Timer B

All timers are stopped after a reset. To restart timers, configure parameters such as operating mode, count source, and counter value, then set the TAiS bit (i = 0 to 4) or TBjS bit (j = 0 to 5) in the TABSR or TBSR register to 1 (count starts).

The following registers and bits should be set while the TAiS bit or TBjS bit is 0 (count stops):

- Registers TAiMR and TBjMR
- The UDF register
- Bits TAZIE, TA0TGL, and TA0TGH in the ONSF register
- The TRGSR register

29.6.2 Timer A

29.6.2.1 Timer Mode

- While the timer counter is running, the TAi register indicates a counter value at any given time. However, FFFFh is read while reloading is in progress. A set value is read if the TAi register is set while the timer counter is stopped.

29.6.2.2 Event Counter Mode

- While the timer counter is running, the TAi register indicates a counter value at any given time. However, FFFFh is read if the timer counter underflows or 0000h if overflows while reloading is in progress. A set value is read if the TAi register is set while the timer counter is stopped.

29.6.2.3 One-shot Timer Mode

- If the TAiS bit in the TABSR register is set to 0 (count stops) while the timer counter is running, the following operations are performed:
  - The timer counter stops and the setting value of the TAi register is reloaded.
  - A low signal is output at the TAiOUT pin.
  - The IR bit in the TAiIC register becomes 1 (interrupts requested) after one CPU clock cycle.

- One-shot timer is operated by an internal count source. When the trigger is an input to the TAiIN pin, the signal is output with a maximum of one count source clock delay after a trigger input to the TAiIN pin.

- The IR bit becomes 1 by any of the settings below. To use the timer Ai interrupt, set the IR bit to 0 after one of the settings below is done:
  - Select one-shot timer mode after a reset.
  - Switch the operating mode from timer mode to one-shot timer mode.
  - Switch the operating mode from event counter mode to one-shot timer mode.

- If a retrigger occurs while counting, the timer counter decrements by one, reloads the setting value of the TAi register, and then continues counting. To generate a retrigger while counting, wait one or more count source cycles after the last trigger is generated.

- When an external trigger input is selected to start counting in timer A one-shot mode, do not provide an external retrigger for 300 ns before the timer counter reaches 0000h. Otherwise, it may stop counting.

29.6.2.4 Pulse-width Modulation Mode

- The IR bit becomes 1 by any of the settings below. To use the timer Ai interrupt (i = 0 to 4), set the IR bit to 0 after one of the settings below is done:
  - Select pulse-width modulation mode after a reset.
  - Switch the operating mode from timer mode to pulse-width modulation mode.
  - Switch the operating mode from event counter mode to pulse-width modulation mode.

- If the TAiS bit in the TABSR register is set to 0 (count stops) while PWM pulse is output, the following operations are performed:
  - The timer counter stops.
  - The output level at the TAiOUT pin changes from high to low. The IR bit becomes 1.
  - When a low signal is output at the TAiOUT pin, it remains unchanged. The IR bit does not change, either.

29.6.3 Timer B

29.6.3.1 Timer Mode and Event Counter Mode

- While the timer counter is running, the TBj register (j = 0 to 5) indicates a counter value at any given time. However, FFFFh is read while reloading is in progress. A set value is read if the TBj register is set while the timer counter is stopped.

29.6.3.2 Pulse Period/Pulse-width Measure Mode

- To set the MR3 bit in the TBjMR register to 0 (no overflow), wait one or more count source cycles to write to the TBjMR register after the MR3 bit becomes 1 (overflow), while the TBjS bit is set to 1 (count starts).

- Use the IR bit in the TBjIC register to detect overflow. The MR3 bit is used only to determine an interrupt request source within the interrupt handler.

- The counter value is undefined when the timer counter starts. Therefore, the timer counter may overflow before a pulse to be measured is applied on the initial valid edge and cause a timer Bj interrupt request to be generated.

- When the pulse to be measured is applied on the initial valid edge after the timer counter starts, an undefined value is transferred to the reload register. At this time, the timer Bj interrupt request is not generated.

- The IR bit may become 1 (interrupt requested) by changing bits MR1 and MR0 in the TBjMR register after the timer counter starts. However, if the same value is rewritten to bits MR1 and MR0, the IR bit is not changed.

- Pulse width is repeatedly measured in pulse-width measure mode. Whether the measurement result is high-level width or not is determined by a program.

- If an overflow occurs simultaneously when a pulse is applied on the valid edge, this pulse is not recognized since an interrupt request is generated only once. Do not let an overflow occur in pulse period measure mode.

- In pulse-width measure mode, determine whether an interrupt source is a pulse applied on the valid edge or an overflow by reading the port level in the TBj interrupt handler.

29.7 Notes on Three-phase Motor Control Timers

29.7.1 Shutdown

- When a low signal is applied to the $\overline{\text{NMI}}$ pin with the bit settings below, pins TA1OUT, TA2OUT, and TA4OUT become high-impedance: the PM24 bit in the PM2 register is 1 (NMI enabled), the INV02 bit in the INVC0 register is 1 (the three-phase motor control timers used), and the INV03 bit is 1 (the three-phase motor control timer output enabled).

29.7.2 Register setting

- Do not write to the TAi1 register (i = 1, 2, 4) in the timing that timer B2 underflows. Before writing to the TAi1 register, read the TB2 register to verify that sufficient time is left until timer B2 underflows. Then, immediately write to the TAi1 register so that no interrupt handler is performed during this write procedure. If the TB2 register indicates little time is left until the underflow, write to the TAi1 register after timer B2 underflows.

29.8 Notes on Serial Interface

29.8.1 Changing the UiBRG Register (i = 0 to 8)

- Set the UiBRG register after setting bits CLK1 and CLK0 in the UiC0 register. When these bits are changed, the UiBRG register must be set again.
- If a clock is input immediately after the UiBRG register is set to 00h, the counter reaches FFh. In this case, it requires an extra 256 clocks to reload 00h into the register. Once the 00h is reloaded, the counter performs the operation without dividing the count source according to the setting.

29.8.2 Synchronous Serial Interface Mode

29.8.2.1 Selecting an External Clock

- If an external clock is selected, the following conditions must be met while the external clock is held high when the CKPOL bit in the UiC0 register (i = 0 to 8) is set to 0 (transmit data output on the falling edge of the transmit/receive clock and receive data input on the rising edge), or while the external clock is held low when the CKPOL bit is set to 1 (transmit data output on the rising edge of the transmit/receive clock and receive data input on the falling edge):
  - The TE bit in the UiC1 register is set to 1 (transmission enabled).
  - The RE bit in the UiC1 register is set to 1 (reception enabled). This bit setting is not required in transmit operation only.
  - The TI bit in the UiC1 register is set to 0 (data held in the UiTB register).

29.8.2.2 Receive Operation

- In synchronous serial interface mode, the transmit/receive clock is controlled by the transmit control circuit. Set the UARTi-associated registers (i = 0 to 8) for a transmit operation, even if the MCU is used only for receive operation. Dummy data is output from the TXDi pin while receiving if the TXDi pin is set to output mode.

- If data is received continuously, an overrun error occurs when the RI bit in the UiC1 register is 1 (data held in the UiRB register) and the seventh bit of the next data is received in the UARTi receive shift register. Then, the OER bit in the UiRB register becomes 1 (overrun error occurred). In this case, the UiRB register becomes undefined. If an overrun error occurs, the IR bit in the SiRIC register is not changed to 1.

29.8.3 Special Mode 1 (I^2C Mode)

- To generate a start condition, stop condition, or restart condition, set the STSPSEL bit in the UiSMR4 register (i = 0 to 6) to 0. Then, wait a half or more clock cycles of the transmit/receive clock to change the respective condition generate bit (the STAREQ, RSTAREQ, or STPREQ bit) from 0 to 1.

29.8.4 Reset Procedure on Communication Error

- Operations which result in communication errors such as rewriting function select registers during transmission/reception should not be performed. Follow the procedure below to reset the internal circuit once the communication error occurs in the following cases: when the operation above is performed by a receiver or transmitter or when a bit slip is caused by noise.

A. Synchronous Serial Interface Mode

(1) Set the TE bit in the UiC1 register (i = 0 to 8) to 0 (transmission disabled) and the RE bit to 0 (reception disabled).
(2) Set bits SMD2 to SMD0 in the UiMR register to 000b (serial interface disabled).
(3) Set bits SMD2 to SMD0 in the UiMR register to 001b (synchronous serial interface mode).
(4) Set the TE bit in the UiC1 register to 1 (transmission enabled) and the RE bit to 1 (reception enabled) if necessary.

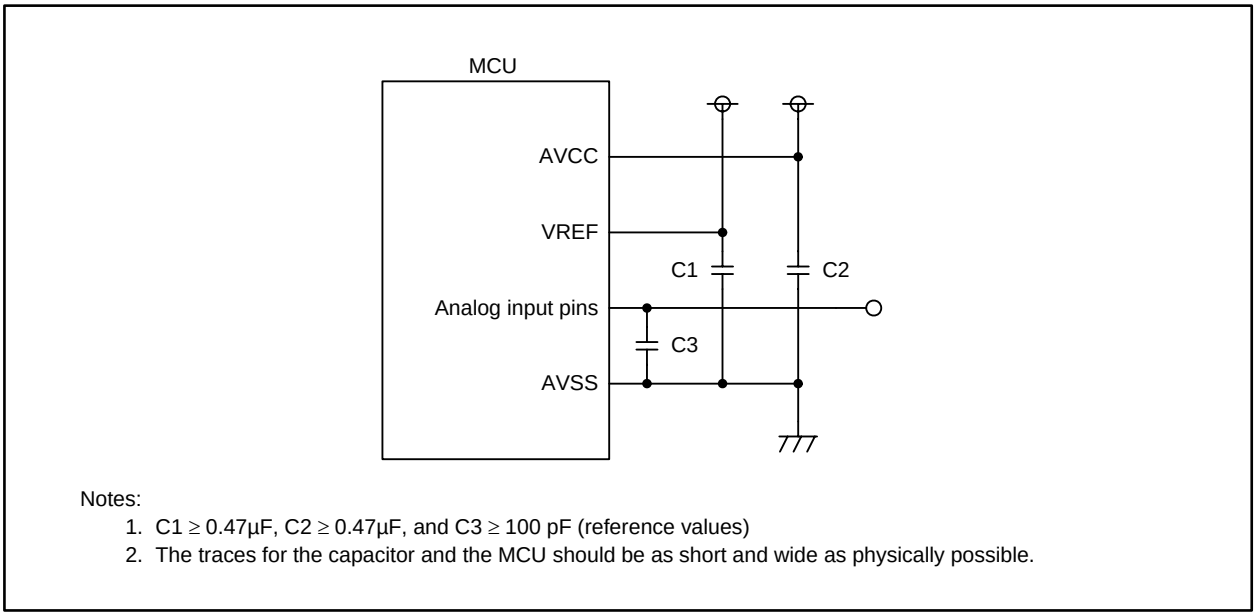
B. UART Mode

(1) Set the TE bit in the UiC1 register to 0 (transmission disabled) and the RE bit to 0 (reception disabled).
(2) Set bits SMD2 to SMD0 in the UiMR register to 000b (serial interface disabled).
(3) Set bits SMD2 to SMD0 in the UiMR register to 100b (UART mode, 7-bit character length), 101b (UART mode, 8-bit character length), or 110b (UART mode, 9-bit character length).
(4) Set the TE bit in the UiC1 register to 1 (transmission enabled) and the RE bit to 1 (reception enabled) if necessary.

29.9 Notes on A/D Converter

29.9.1 Notes on Designing Boards

- Three capacitors should be respectively placed between the AVSS pin and such pins as AVCC, VREF, and analog inputs (AN_0 to AN_7, AN0_0 to AN0_7, AN2_0 to AN2_7, and AN15_0 to AN15_3) to avoid error operations caused by noise or latchup, and to reduce conversion errors. Figure 29.1 shows an example of pin configuration for A/D converter.



Notes:
1. C1 ≥ 0.47µF, C2 ≥ 0.47µF, and C3 ≥ 100 pF (reference values)
2. The traces for the capacitor and the MCU should be as short and wide as physically possible.

Figure 29.1 Pin Configuration for A/D Converter

- Do not use any of the four pins AN_4 to AN_7 for analog input if the key input interrupt is to be used. Otherwise, a key input interrupt request occurs when the A/D input voltage becomes VIL or lower.

- When AVCC = VREF = VCC, A/D input voltage for pins AN_0 to AN_7, AN0_0 to AN0_7, AN2_0 to AN2_7, AN15_0 to AN15_3, ANEX0, and ANEX1 should be VCC or lower.

29.9.2 Notes on Programming

- The following registers should be written while the A/D conversion is stopped, that is, before a trigger occurs: AD0CON0 (except the ADST bit), AD0CON1, AD0CON2, AD0CON3, and AD0CON4.

- If the VCUT bit in the AD0CON1 register is switched from 0 (VREF connected) to 1 (VREF disconnected), the A/D conversion should be started after 1 µs or more. Set the VCUT bit to 0 when A/D conversion is not used to reduce power consumption.

- Set the port direction bit for the pin to be used as an analog input pin to 0 (input). Set the ASEL bit of the corresponding port function select register to 1 (the port is used as A/D input).

- If the TRG bit in the AD0CON0 register is set to 1 (external trigger or hardware trigger is selected), set the corresponding port direction bit (PD9_7 bit) for the $\overline{\text{ADTRG}}$ pin to 0 (input).

- The ϕAD frequency should be 10 MHz or below. It should be 1 MHz or above if the sample and hold function is enabled. If not, it should be 250 kHz or above.

- If A/D operating mode (bits MD1 and MD0 in the AD0CON0 register or the MD2 bit in the AD0CON1 register) has been changed, re-select analog input pins by using bits CH2 to CH0 in the AD0CON0 register or bits SCAN1 and SCAN0 in the AD0CON1 register.

- If the AD0i register (i = 0 to 7) is read when the A/D converted result is stored to the register, the stored value may have an error. Read the AD0i register after the A/D conversion has been completed.
  In one-shot mode or single sweep mode, read the respective AD0i register after the IR bit in the AD0IC register has become 1 (interrupt requested).
  In repeat mode, repeat sweep mode 0, or repeat sweep mode 1, an interrupt request can be generated each time when an A/D conversion has been completed if the DUS bit in the AD0CON3 register is set to 1 (DMAC operating mode enabled). Similar to the other modes above, read the AD00 register after the IR bit in the AD0IC register has become 1 (interrupt requested).

- If the A/D conversion in progress is halted by setting the ADST bit in the AD0CON0 register to 0, the converted result is undefined. In addition, the unconverted AD0i register may also become undefined. Consequently, the AD0i register should not be used just after A/D conversion is halted.

- The external trigger cannot be used in DMAC operating mode. When the DMAC is configured to transfer converted results, do not read the AD00 register by a program.

- If, in single sweep mode, the A/D conversion in progress is halted by setting the ADST bit in the AD0CON0 register to 0 (A/D conversion is stopped), an interrupt request may be generated even though the sweep is not completed. To halt the A/D conversion, first disable interrupts, then set the ADST bit to 0.

29.10 Notes on Flash Memory Rewriting

29.10.1 Note on Power Supply

- Keep the supply voltage constant within the range specified in the electrical characteristics while a rewrite operation on flash memory is in progress. If the supply voltage becomes beyond the guaranteed value, the device cannot be guaranteed.

29.10.2 Note on Hardware Reset

- Do not perform a hardware reset while a rewrite operation on flash memory is in progress.

29.10.3 Note on Flash Memory Protection

- If an ID code written in an assigned address has an error, any read/write operation of flash memory in standard serial I/O mode is disabled.

29.10.4 Notes on Programming

- Do not set the FEW bit in the FMCR register to 1 (CPU rewrite mode) in low speed mode or low power mode.
- Four software commands of program, block erase, lock bit program, and protect bit program are interrupted by an NMI, a watchdog timer interrupt, or an oscillator stop detection interrupt. If any of the software commands above is interrupted, erase the corresponding block and then execute the same command again. If the block erase command is interrupted, values of lock bits and protect bits become undefined. Therefore, disable the lock bit, and then execute the block erase command again.

29.10.5 Notes on Interrupts

- EW0 mode
  - To use interrupts assigned to the relocatable vector table, the vector table should be addressed in RAM space.
  - If either of an NMI, a watchdog timer interrupt, or an oscillator stop detection interrupt is generated, the flash memory module automatically enters read array mode. Therefore these interrupts are enabled even during a rewrite operation. On the other hand, the rewrite operation in progress is aborted by the interrupt and registers FMR0 and FRSR0 are reset. When the interrupt handler has ended, set the LBD bit in the FMR1 register to 1 (lock bit protection disabled) to re-execute the rewrite operation.
  - Instructions BRK, INTO, and UND, which refer to data on the flash memory, are unavailable in this mode.

- EW1 mode
  - Interrupts assigned to the relocatable vector table should not be accepted during a program or block erase operation.
  - The watchdog timer interrupt should not be generated, either.
  - If either of an NMI, a watchdog timer interrupt, or an oscillator stop detection interrupt is generated, the flash memory module automatically enters read array mode. Therefore this interrupt is enabled even during a rewrite operation. On the other hand, the rewrite operation in progress is aborted by the interrupt and registers FMR0 and FRSR0 are reset. When the interrupt handler has ended, set the EWM bit in the FMR0 register to 1 (set as EW1 mode) and the LBD bit in the FMR1 register to 1 (lock bit protection disabled) to re-execute the rewrite operation.

29.10.6 Notes on Rewrite Control Program

- EW0 mode
  - If the supply voltage lowers during the rewrite operation of blocks having the rewrite control program, the rewrite control program may not be successfully rewritten, then the rewrite operation itself may not be performed. In this case perform the rewrite operation by serial programmer or parallel programmer.

- EW1 mode
  - Do not rewrite blocks having the rewrite control program.

29.10.7 Notes on Number of Programming/Erasure and Software Command Execution Time

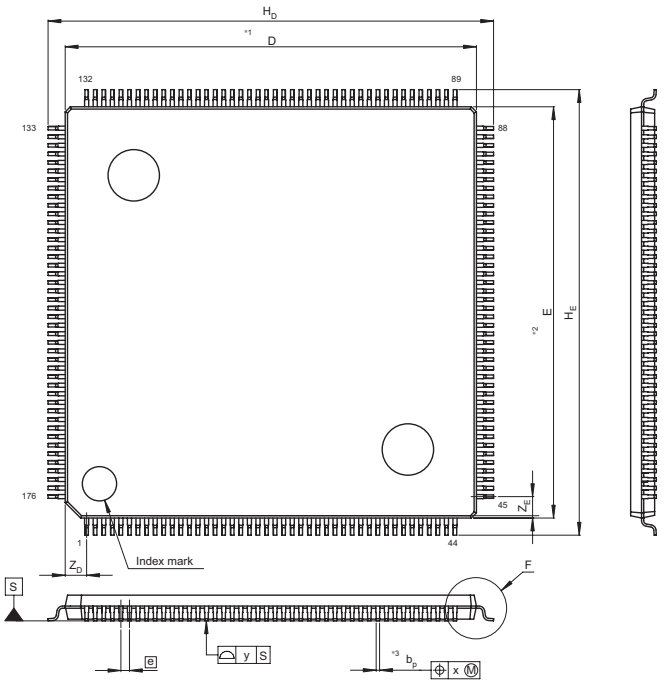
- According to the increase of program/erase operation, the four software commands: program, block erase, lock bit program, and protect bit program require more time to be executed. If the number of programming/erasure exceeds the minimum endurance value specified in the electrical characteristics, it may take unpredictable time to execute the software commands. The waiting time for the execution of software commands should be set much longer than the execution time specified in the electrical characteristics.

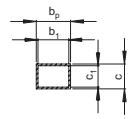
29.10.8 Other Notes

- The required time to perform the program or erase operation specified in the electrical characteristics can be guaranteed within the minimum values of programming/erasure endurance specified in the same table. Even if the number of programming/erasure exceeds the minimum endurance value, the program or erase operation may be unguaranteedly performed.
- Chips repeatedly programmed and erased for debugging are not allowed to be used for commercial products.

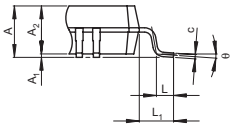
Appendix 1. Package Dimensions

JEITA Package Code	RENESAS Code	Previous Code	MASS[Typ.]
P-LQFP176-24x24-0.50	PLQP0176KB-A	176P6Q-A/FP-176E/FP-176EV	1.8g





Terminal cross section



Detail F

NOTE)
1. DIMENSIONS "*1" AND "*2" DO NOT INCLUDE MOLD FLASH.
2. DIMENSION "*3" DOES NOT INCLUDE TRIM OFFSET.

Reference Symbol	Dimension in Millimeters		
	Min	Nom	Max
D	23.9	24.0	24.1
E	23.9	24.0	24.1
A_2	—	1.4	—
H_D	25.8	26.0	26.2
H_E	25.8	26.0	26.2
A	—	—	1.7
A_1	0.05	0.1	0.15
b_p	0.15	0.20	0.25
b_1	—	0.18	—
c	0.09	0.145	0.20
c_1	—	0.125	—
θ	0°	—	8°
e	—	0.5	—
x	—	—	0.08
y	—	—	0.10
Z_D	—	1.25	—
Z_E	—	1.25	—
L	0.35	0.5	0.65
L_1	—	1.0	—

INDEX

A

A0 17
A1 17
A2 17
A3 17
AD00 to AD07 266
AD0CON0 262
AD0CON1 263
AD0CON2 264
AD0CON3 265
AD0CON4 266
AD0IC 111
Address Register 17
AIFC 507

B

B Flag 18
BCN0IC to BCN6IC 111
BRK Instruction Interrupt 102
BRK2 Instruction Interrupt 102
BTSR 297

C

C Flag 17
C0AFSR 404
C0BCR 376
C0CLKR 375
C0CSSR 403
C0CTLR 371
C0ECSR 412
C0EIC 111
C0EIER 405
C0EIFR 407
C0FIDCR0, C0FIDCR1 379
C0FRIC 111
C0FTIC 111
C0MB0 to C0MB31 383
C0MCTL0 to C0MCTL31 387
C0MIER 386
C0MKIVLR 381
C0MKR0 to C0MKR7 378
C0MSMR 400
C0MSSR 401
C0RECR 410
C0RFCR 390
C0RFPCR 393
C0RIC 111
C0STR 397
C0TCR 415
C0TECR 411
C0TFCR 394
C0TFPCR 396
C0TIC 111
C0TSR 414
C0WIC 111
CADDS 511
Carry Flag 17
CCR 68
CDADS 511, 512
CDCR 506
CDSC 511
CM0 69
CM1 70
CM2 71
CM3 71
CPSRF 72
CRCD 282
CRCIN 283

D

D Flag 17
DA0, DA1 281
DACON 281
Data Register 17
DCKC 520
DCR0 to DCR3 ..... 19, 136
DCT0 to DCT3 ..... 19, 135
DDA0 to DDA3 ..... 19, 137
DDR0 to DDR3 ..... 19, 137
Debug Flag 17
DIC0 518
DIC1 519
DM0IC to DM3IC 111
DM0SL to DM3SL 131
DM0SL2 to DM3SL2 132
DMA Destination Address Register 19
DMA Destination Address Reload Register 19
DMA Mode Register 19
DMA Source Address Register 19
DMA Source Address Reload Register 19
DMA Terminal Count Register 19
DMA Terminal Count Reload Register 19
DMD0 to DMD3 ..... 19, 135
DP Bit 18
DPC0 512
DPC1 513
DPC10 518
DPC2 513
DPC3 514

DPC4 514
DPC5 515
DPC6 516
DPC7 517
DPC8 517
DPLC 508
DPRR 519
DSA0 to DSA3 ..... 19, 136
DSPC 509
DSR0 to DSR3 ..... 19, 137
DTT 203
DVRC 510

F

Fast Interrupt 104
FB 17
FBPM0 476
FBPM1 476
FBPM2 477
FEBC 473
Fixed-point Designation Bit 18
Flag Register 17
FLG 17
Floating-point Overflow Flag 18
Floating-point Rounding Mode 18
Floating-point Underflow Flag 18
FMCR 472
FMR0 474
FMR1 475
FMSR0 475
FO Flag 18
FPR0 474
Frame Base Register 17
FU Flag 18

G

G0BCR0 to G2BCR0 294
G0BCR1, G1BCR1 295
G0BT to G2BT 293
G0FE to G2FE 303
G0FS, G1FS 302
G0PO0 to G0PO7 302
G0POCR0 to G0POCR7 300
G0TM0 to G0TM7 299
G0TMCR0 to G0TMCR7 298
G0TPR6, G0TPR7 298
G1PO0 to G1PO7 302
G1POCR0 to G1POCR7 300
G1TM0 to G1TM7 299
G1TMCR0 to G1TMCR7 298
G1TPR6, G1TPR7 298
G2BCR1 296
G2CR 330
G2MR 329
G2PO0 to G2PO7 302
G2POCR0 to G2POCR7 301
G2RB 329
G2RTP 303
G2TB 328

H

Hardware Interrupt 103

I

I Flag 18
I2CCCR 342
I2CCR0 340
I2CCR1 345
I2CCR2 348
I2CIC 111
I2CLIC 111
I2CMR 354
I2CSAR 339
I2CSR 350
I2CSSCR 344
I2CTRSR 338
ICTB2 195
IDB0, IDB1 194
IEAR 331
IECR 330
IERIF 332
IETIF 331
IFS0 456
IFS1 457
IFS2 458
IFS3 459
IFSR0 ..... 120, 222
IFSR1 ..... 121, 223
IIO0IC to IIO11IC 111
IIO0IE to IIO11IE 125
IIO0IR to IIO11IR 124
INT Instruction Interrupt 102
INT0IC to INT8IC 112
INTB 17
Interrupt Control Register 111
Interrupt Enable Flag 18
Interrupt request level 112
Interrupt Response Time 116
Interrupt Sequence 115
Interrupt Stack Pointer 17
Interrupt Types 101
Interrupt Vector Table Base Register 17

INVC0 191
INVC1 192
IOBC 193
IPL 18, 110
ISP 17

K

KUPIC 111

M

Maskable Interrupt 101
MOD 148

N

NMI (Non Maskable Interrupt) 103
Non-maskable Interrupt 101

O

O Flag 18
ONSF 162
Oscillator Stop Detection Interrupt 103
Overflow Flag 18
Overflow Interrupt 102

P

P0 to P15 155
P0_0S to P0_7S 439
P10_0S to P10_7S 450
P1_0S to P1_7S 440
P11_0S to P11_4S 451
P11_0S to P11_7S 451
P12_0S to P12_7S 452
P13_0S to P13_7S 453
P14_3S to P14_6S 454
P15_0S to P15_7S 455
P2_0S to P2_7S 441
P3_0S to P3_7S 442
P4_0S to P4_7S 443
P5_0S to P5_7S 444
P6_0S to P6_7S 445
P7_0S to P7_7S 446
P8_0S to P8_4S, P8_6S, P8_7S 447
P9_0S to P9_7S 448
PBC 97
PC 17
PCR 463
PD0 to PD15 437
Peripheral Interrupt 103
PLC0 77
PLC1 78
PM0 62
PM2 73
PM3 74
PRCR 98
PRCR2 99
PRCR3 99
Processor Interrupt Priority Level 18, 110
Program Counter 17
PRR 100
PUR0 460
PUR1 460
PUR2 461
PUR3 461
PUR4 462

R

R2R0 17
R3R1 17
R6R4 17
R7R5 17
Register Bank Select Flag 18
Register Saving 117
RIPL1, RIPL2 ..... 114, 145
RND 18

S

S Flag 17
S0RIC to S8RIC 111
S0TIC to S8TIC 111
Save Flag Register 19
Save PC Register 19
SB 17
Sign Flag 17
Single-step Interrupt 103
Software Interrupt 102
SP 17
Special Interrupt 103
Stack Pointer 17
Stack Pointer Select Flag 18
Static Base Register 17
SVF 19
SVP 19

T

TA0 to TA4 159
TA0IC to TA4IC 111
TA0MR to TA4MR .. 160, 166, 169, 172, 174

TA1, TA2, TA4, TA11, TA21, TA41 199
TA1MR, TA2MR, TA4MR 200
TABSR 160, 178, 201
TB0 to TB5 177
TB0IC to TB5IC 111
TB0MR to TB5MR 177, 180, 182, 184
TB2 197
TB2MR 197
TB2SC 198
TBSR 178
TCSPR 72, 164
TRGSR 163, 201

U

U Flag 18
U0BRG to U8BRG 219
U0C0 to U6C0 212
U0C1 to U6C1 214
U0MR to U6MR 210
U0RB to U6RB 220
U0SMR to U6SMR 216
U0SMR2 to U6SMR2 217
U0SMR3 to U6SMR3 218
U0SMR4 to U6SMR4 219
U0TB to U8TB 220
U78CON 215
U7C0, U8C0 213
U7C1, U8C1 214
U7MR, U8MR 211
U7RB, U8RB 221
UDF 161
Undefined Instruction Interrupt 102
User Stack Pointer 17
USP 17

V

VCT 19
Vector Register 19
VRCR 64

W

Watchdog Timer Interrupt 103
WDC 128
WDTS 128

X

X0R to X15R 285
XYC 285

Y

Y0R to Y15R 286

Z

Z Flag 17
Zero Flag 17

Revision History	R32C/102 Group User's Manual: Hardware

Rev.	Date	Description	
		Page	Summary
1.01	Nov 15, 2010	—	Initial release

R32C/102 Group User’s Manual: Hardware

Publication Date: Rev. 1.01 Nov 15, 2010

Published by: Renesas Electronics Corporation

SALES OFFICES Renesas Electronics Corporation http://www.renesas.com

Refer to "http://www.renesas.com/" for the latest and detailed information.

Renesas Electronics America Inc.
2880 Scott Boulevard Santa Clara, CA 95050-2554, U.S.A.
Tel: +1-408-588-6000, Fax: +1-408-588-6130

Renesas Electronics Canada Limited
1101 Nicholson Road, Newmarket, Ontario L3Y 9C3, Canada
Tel: +1-905-898-5441, Fax: +1-905-898-3220

Renesas Electronics Europe Limited
Dukes Meadow, Millboard Road, Bourne End, Buckinghamshire, SL8 5FH, U.K
Tel: +44-1628-585-100, Fax: +44-1628-585-900

Renesas Electronics Europe GmbH
Arcadiastrasse 10, 40472 Düsseldorf, Germany
Tel: +49-211-65030, Fax: +49-211-6503-1327

Renesas Electronics (China) Co., Ltd.
7th Floor, Quantum Plaza, No.27 ZhiChunLu Haidian District, Beijing 100083, P.R.China
Tel: +86-10-8235-1155, Fax: +86-10-8235-7679

Renesas Electronics (Shanghai) Co., Ltd.
Unit 204, 205, AZIA Center, No.1233 Lujiazui Ring Rd., Pudong District, Shanghai 200120, China
Tel: +86-21-5877-1818, Fax: +86-21-6887-7858 / -7898

Renesas Electronics Hong Kong Limited
Unit 1601-1613, 16/F., Tower 2, Grand Century Place, 193 Prince Edward Road West, Mongkok, Kowloon, Hong Kong
Tel: +852-2886-9318, Fax: +852 2886-9022/9044

Renesas Electronics Taiwan Co., Ltd.
7F, No. 363 Fu Shing North Road Taipei, Taiwan
Tel: +886-2-8175-9600, Fax: +886 2-8175-9670

Renesas Electronics Singapore Pte. Ltd.
1 harbourFront Avenue, #06-10, keppel Bay Tower, Singapore 098632
Tel: +65-6213-0200, Fax: +65-6278-8001

Renesas Electronics Malaysia Sdn.Bhd.
Unit 906, Block B, Menara Amcorp, Amcorp Trade Centre, No. 18, Jln Persiaran Barat, 46050 Petaling Jaya, Selangor Darul Ehsan, Malaysia
Tel: +60-3-7955-9390, Fax: +60-3-7955-9510

Renesas Electronics Korea Co., Ltd.
11F., Samik Lavied' or Bldg., 720-2 Yeoksam-Dong, Kangnam-Ku, Seoul 135-080, Korea
Tel: +82-2-558-3737, Fax: +82-2-558-5141

© 2010 Renesas Electronics Corporation. All rights reserved.

Colophon 1.0

R32C/102 Group

Renesas Electronics Corporation

REJ09B0578-0101